

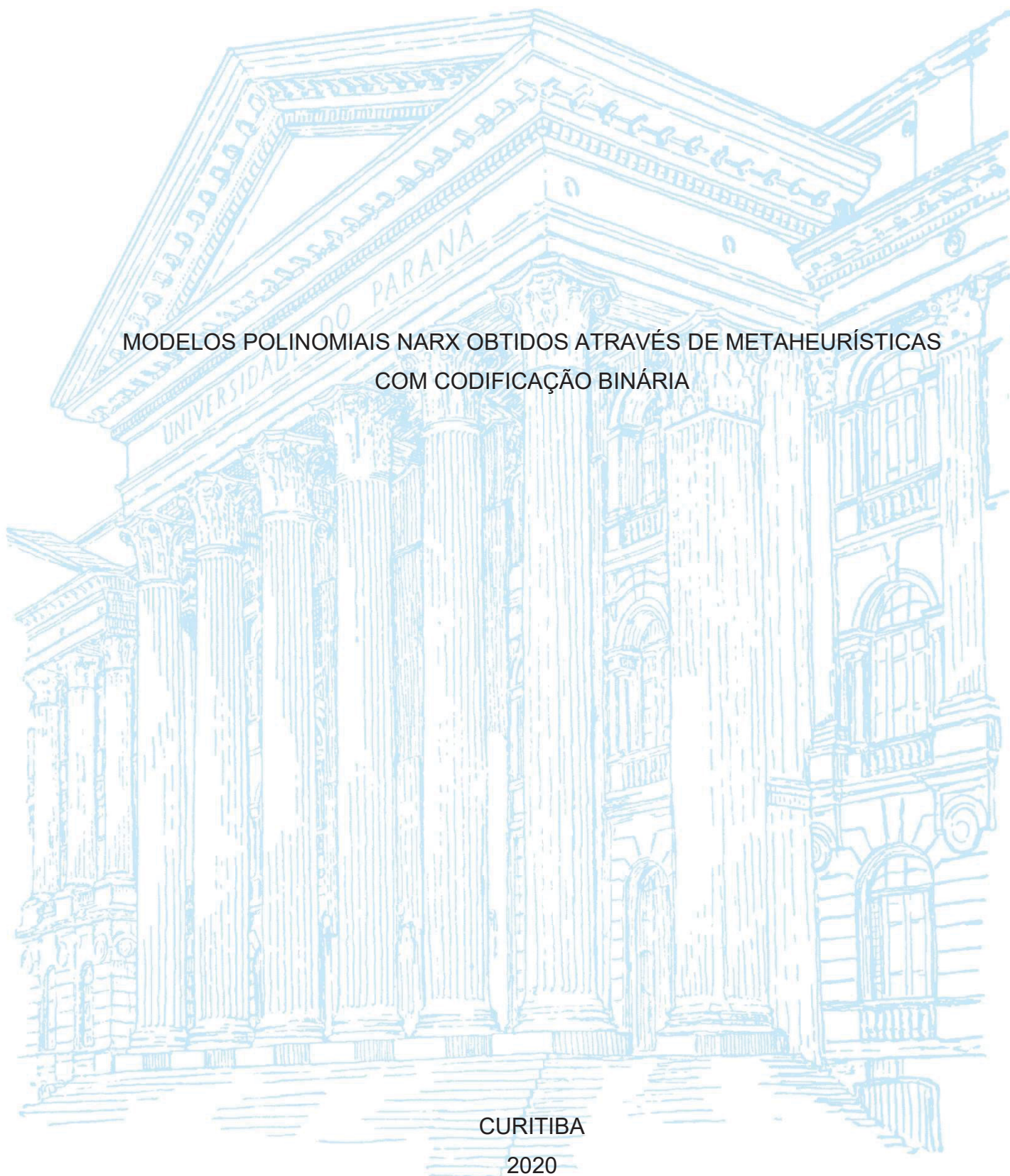
UNIVERSIDADE FEDERAL DO PARANÁ

ÁDAMO HENRIQUE ROCHA DE OLIVEIRA

MODELOS POLINOMIAIS NARX OBTIDOS ATRAVÉS DE METAHEURÍSTICAS  
COM CODIFICAÇÃO BINÁRIA

CURITIBA

2020



ÁDAMO HENRIQUE ROCHA DE OLIVEIRA

MODELOS POLINOMIAIS NARX OBTIDOS ATRAVÉS DE METAHEURÍSTICAS  
COM CODIFICAÇÃO BINÁRIA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Gideon Villar Leandro

CURITIBA

2020

CATALOGAÇÃO NA FONTE – SIBI/UFPR

---

O48m

Oliveira, Ádamo Henrique Rocha de

Modelos polinomiais narx obtidos através de metaheurísticas com codificação binária [recurso eletrônico]/ Ádamo Henrique Rocha de Oliveira, 2020.

Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Orientador: Prof. Dr. Gideon Villar Leandro

1. Engenharia elétrica. I. Leandro, Gideon Villar. II. Universidade Federal do Paraná. III. Título.

CDD 621.3

---

Bibliotecária: Vilma Machado CRB9/1563

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **ÁDAMO HENRIQUE ROCHA DE OLIVEIRA** intitulada: **MODELOS POLINOMIAIS NARX OBTIDOS ATRAVÉS DE METAHEURÍSTICAS COM CODIFICAÇÃO BINÁRIA**, sob orientação do Prof. Dr. GIDEON VILLAR LEANDRO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 24 de Junho de 2020.

Assinatura Eletrônica

25/06/2020 17:35:32.0

GIDEON VILLAR LEANDRO

Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica

25/06/2020 16:51:55.0

HELON VICENTE HULTMANN AYALA

Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA - PUC/RIO)

Assinatura Eletrônica

25/06/2020 17:51:53.0

LEANDRO DOS SANTOS COELHO

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)



*A todos aqueles que, através da educação, comprometem-se a tornar  
o mundo um lugar cada vez melhor.*

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por toda força e sabedoria concedidas ao longo da minha jornada pessoal e profissional.

Agradeço à minha família por todo o apoio que sempre me ofereceram, principalmente minha mãe, Cláudia, por ser um exemplo de pessoa batalhadora.

Ao Instituto Federal do Maranhão pela concessão da licença que me permitiu realizar os estudos de mestrado.

Aos meus amigos Carlos, Juliana, Karine e Yrles por todas as conversas, conselhos e momentos de suporte.

A todos os integrantes do Lasico, que se tornaram minha família em Curitiba, principalmente Mariane, Carla, Cristiano, Wagner e Douglas

A todos os professores e funcionários da UFPR pelas colaborações diretas e indiretas na realização do meu mestrado.

Aos professores Dr. Gustavo Henrique da Costa Oliveira, Dr. Helon Vicente Hultmann Ayala e Dr. Leandro dos Santos Coelho pelas sugestões de melhoria deste trabalho.

E agradeço especialmente ao meu orientador, prof. Dr. Gideon Villar Leandro, por todo direcionamento e suporte oferecidos durante o mestrado, pela paciência e dedicação no decorrer da orientação, e pelas valiosas contribuições que levarei para a vida pessoal e profissional.

*“O homem que move montanhas começa carregando pequenas pedras.”*

(Provérbio chinês)

## RESUMO

Dentro do contexto da identificação de sistemas, a etapa de seleção de estrutura representa uma tarefa complexa por se tratar de um problema de otimização combinatória do tipo binária. Para solucionar este problema, diversas técnicas vêm sendo aplicadas, dentre elas a seleção por meio de metaheurísticas. Entretanto, devido à grande diversidade de metaheurísticas existentes na literatura, a escolha daquela mais adequada para cumprir esta tarefa pode ser algo desafiador para o projetista. Neste trabalho, é realizada uma análise do comportamento de metaheurísticas aplicadas ao problema de seleção de estrutura, além de ser apresentado um novo método de codificação binária, chamado de Modulação em Ângulo Modificada (MAM), que tende a melhorar o desempenho das metaheurísticas neste tipo de problema. Normalmente as metaheurísticas atuam na seleção de estrutura originalmente manipulando soluções binárias ou através de associação com alguma técnica de codificação binária. Foram avaliados os desempenhos das metaheurísticas Algoritmo Genético, Evolução Diferencial e Algoritmo do Morcego aplicadas ao problema de seleção de estrutura de modelos não lineares autorregressivos com entrada externa (NARX, do inglês *Nonlinear AutoRegressive with eXogenous input*). O Algoritmo Genético já consiste em uma metaheurística projetada para manipular soluções binárias. A Evolução Diferencial e o Algoritmo do Morcego, por sua vez, tiveram suas versões binárias implementadas através das codificações Função de Transferência (TF), Prioridade de Maior Valor (GVP) e Modulação em Ângulo (AM). Além disso, a forma de associação entre as metaheurísticas e a codificação AM foi modificada, dando origem à codificação MAM. Dois estudos de caso foram conduzidos utilizando dados de um conversor *buck* e de um aquecedor elétrico. Os resultados das simulações mostram que as versões binárias da Evolução Diferencial obtidas com as codificações TF, GVP e MAM foram as que apresentaram melhores desempenho, superando o Algoritmo Genético e o Algoritmo do Morcego. Além disso, levando em conta a convergência das soluções e a capacidade de localizar bons modelos, em todos os cenários analisados o desempenho da Evolução Diferencial codificada com MAM melhorou substancialmente em relação à sua versão original codificada com AM. Os melhores modelos encontrados neste trabalho apresentaram bom desempenho ao serem aplicados métodos de validação (simulação livre e análise de resíduos) e ao serem comparados com modelos da literatura.

Palavras-chave: Identificação de sistemas. Seleção de estrutura. Modelo NARX. Metaheurísticas. Codificação binária.

## ABSTRACT

During a system identification procedure, structure selection represents a complex task because it is a binary combinatorial optimization problem. To solve this problem, several techniques have been applied, the metaheuristics is one of them. However, there is a great diversity of metaheuristics, thus choosing the most suitable one to perform the task is difficult for the designer. In this study, we performed a performance analysis of metaheuristics applied to a structure selection problem. In addition, we present a new binarization technique, called Modified Angle Modulation (MAM), which tends to improve the performance of metaheuristics. Usually, metaheuristics perform the structure selection taking binary solutions directly or through the association with a binarization technique. We evaluated the performances of three metaheuristic techniques, being the Genetic Algorithm, Differential Evolution and Bat Algorithm, all working at a structure selection problem for nonlinear autoregressive models with exogenous input (NARX). The Genetic Algorithm is originally a binary metaheuristics. Binary versions of the Differential Evolution and the Bat Algorithm were developed through the Transfer Function (TF), Great Value Priority (GVP) and Angle Modulation (AM) binarizations. In addition, the form of association between metaheuristics and AM binarization has been modified, originating the MAM binarization. We conducted two case studies using data from a buck converter and an electric heater. Binary versions of the Differential Evolution developed through TF, GVP and MAM binarizations performed better than the Genetic Algorithm and Bat Algorithm. Furthermore, considering the convergence of solutions and the ability to locate good models, the performance of the binary version of the Differential Evolution developed with MAM substantially improved in relation to its original version developed with AM. Finally, the best estimated models performed well not only during validation tests (free run simulation and statistical validation), but also when compared with other models available in the literature.

Keywords: System identification. Structure selection. NARX models. Metaheuristics. Binarization.



## LISTA DE FIGURAS

FIGURA 1 – Escala com diferentes níveis de informação auxiliar na identificação caixa cinza .....	22
FIGURA 2 – Diagrama do processo de predição de um passo à frente.....	34
FIGURA 3 – Diagrama do processo de simulação livre .....	35
FIGURA 4 – Cromossomo em que cada bit representa um gene .....	41
FIGURA 5 – Cruzamento de um único ponto .....	44
FIGURA 6 – Cruzamento de dois pontos .....	45
FIGURA 7 – Função trigonométrica para Modulação em Ângulo.....	57
FIGURA 8 – Fluxograma do procedimento aplicado para seleção de estrutura e obtenção de modelos.....	61
FIGURA 9 – Função utilizada na codificação TF.....	62
FIGURA 10 – Função geradora utilizada na codificação MAM .....	63
FIGURA 11 – Exemplo de curva de convergência média .....	66
FIGURA 12 – Diagrama do conversor <i>buck</i> .....	68
FIGURA 13 – Dados coletados do conversor <i>buck</i> . (a) sinal de entrada e (b) sinal de saída. ....	69
FIGURA 14 – Sinais de entrada e saída do conversor <i>buck</i> . Dados de estimação (a) e validação (b).....	70
FIGURA 15 – Conjunto de termos candidatos para geração de modelos do conversor <i>buck</i> .....	71
FIGURA 16 – Convergência média para o GA (Estudo de caso 1).....	73
FIGURA 17 – Convergência média para BA e DE codificados com TF (Estudo de caso 1) .....	75
FIGURA 18 – Convergência média para BA e DE codificados com GVP (Estudo de caso 1) .....	77
FIGURA 19 – Convergência média para BA e DE codificados com AM (Estudo de caso 1) .....	78
FIGURA 20 – Convergência média para BA e DE codificados com MAM (Estudo de caso 1) .....	80
FIGURA 21 – Simulação livre dos modelos CB01 a CB07 .....	83
FIGURA 22 – Validação estatística do modelo CB01. ....	85
FIGURA 23 – Validação estatística do modelo CB02. ....	85

FIGURA 24 – Validação estatística do modelo CB03. ....	86
FIGURA 25 – Validação estatística do modelo CB04. ....	86
FIGURA 26 – Validação estatística do modelo CB05. ....	87
FIGURA 27 – Validação estatística do modelo CB06. ....	87
FIGURA 28 – Validação estatística do modelo CB07. ....	88
FIGURA 29 – Análise dos regressores dos modelos CB01 a CB07. ....	89
FIGURA 30 – Simulação livre dos modelos CB01, CB06 e Aguirre .....	90
FIGURA 31 – Simulação livre dos modelos CB08 e Aguirre .....	92
FIGURA 32 – Simulação livre dos modelos CB09 e Aguirre .....	93
FIGURA 33 – Validação estatística do modelo CB09. ....	94
FIGURA 34 – Diagrama de blocos do aquecedor elétrico .....	95
FIGURA 35 – Dados coletados do aquecedor elétrico. (a) sinal de entrada e (b) sinal de saída. ....	96
FIGURA 36 – Sinais de entrada e saída do aquecedor elétrico. Dados de estimação (a) e validação (b). ....	97
FIGURA 37 – Conjunto de termos candidatos para geração de modelos do aquecedor elétrico.....	98
FIGURA 38 – Convergência média para o GA (Estudo de caso 2).....	99
FIGURA 39 – Convergência média para BA e DE codificados com TF (Estudo de caso 2) .....	101
FIGURA 40 – Convergência média para BA e DE codificados com GVP (Estudo de caso 2) .....	102
FIGURA 41 – Convergência média para BA e DE codificados com AM (Estudo de caso 2) .....	104
FIGURA 42 – Convergência média para BA e DE codificados com MAM (Estudo de caso 2) .....	107
FIGURA 43 – Simulação livre dos modelos AE01 a AE10 .....	112
FIGURA 44 – Validação estatística do modelo AE01.....	113
FIGURA 45 – Validação estatística do modelo AE02.....	114
FIGURA 46 – Validação estatística do modelo AE03.....	114
FIGURA 47 – Validação estatística do modelo AE04.....	115
FIGURA 48 – Validação estatística do modelo AE05.....	115
FIGURA 49 – Validação estatística do modelo AE06.....	116
FIGURA 50 – Validação estatística do modelo AE07.....	116

FIGURA 51 – Validação estatística do modelo AE08.....	117
FIGURA 52 – Validação estatística do modelo AE09.....	117
FIGURA 53 – Validação estatística do modelo AE10.....	118
FIGURA 54 – Análise dos regressores dos modelos AE01 a AE10.....	119
FIGURA 55 – Simulação livre dos modelos AE08 e Rodrigues (início: $k = 20$ ).....	121
FIGURA 56 – Simulação livre dos modelos AE08 e Rodrigues (início: $k = 3$ ).....	122

## LISTA DE TABELAS

TABELA 1 – Estratégias da Evolução Diferencial .....	50
TABELA 2 – Exemplo de aplicação da codificação binária utilizando GVP .....	56
TABELA 3 – Abordagens utilizadas pelo Algoritmo Genético .....	65
TABELA 4 – Técnicas aplicadas no algoritmo de seleção de estrutura .....	65
TABELA 5 – Parâmetros das metaheurísticas (Estudo de caso 1) .....	72
TABELA 6 – Valores de RMSE alcançados pelo GA (Estudo de caso 1) .....	74
TABELA 7 – Valores de RMSE alcançados por BA e DE com codificação TF .....	76
TABELA 8 – Valores de RMSE alcançados por BA e DE com codificação GVP (Estudo de caso 1) .....	77
TABELA 9 – Valores de RMSE alcançados por BA e DE com codificação AM (Estudo de caso 1) .....	78
TABELA 10 – Valores de RMSE alcançados por BA e DE com codificação MAM (Estudo de caso 1) .....	80
TABELA 11 – Taxa de localização das técnicas aplicadas na seleção de estrutura (Estudo de caso 1) .....	81
TABELA 12 – Custo computacional de cada técnica utilizada (Estudo de caso 1) ...	82
TABELA 13 – Índices de desempenho para os modelos CB01 a CB07 .....	84
TABELA 14 – Índices de desempenho para os modelos CB01, CB06 e Aguirre .....	91
TABELA 15 – Índices de desempenho para o modelo CB08 .....	92
TABELA 16 – Índices de desempenho para o modelo CB09 .....	93
TABELA 17 – Parâmetros das metaheurísticas (Estudo de caso 2) .....	99
TABELA 18 – Valores de RMSE alcançados pelo GA (Estudo de caso 2) .....	100
TABELA 19 – Valores de RMSE alcançados por BA e DE com codificação TF (Estudo de caso 2) .....	102
TABELA 20 – Valores de RMSE alcançados por BA e DE com codificação GVP (Estudo de caso 2) .....	103
TABELA 21 – Valores de RMSE alcançados por BA e DE com codificação AM (Estudo de caso 2) .....	105
TABELA 22 – Valores de RMSE alcançados por BA e DE com codificação MAM (Estudo de caso 2) .....	107
TABELA 23 – Taxa de localização das técnicas aplicadas na seleção de estrutura (Estudo de caso 2) .....	109

TABELA 24 – Custo computacional de cada técnica utilizada (Estudo de caso 2) .	110
TABELA 25 – Índices de desempenho para os modelos AE01 a AE10.....	113
TABELA 26 – Índices de desempenho para os modelos AE08 e Rodrigues (início: $k$ = 20).....	121
TABELA 27 – Índices de desempenho para os modelos AE08 e Rodrigues (início: $k$ = 3).....	122



## LISTA DE ABREVIATURAS OU SIGLAS

AIC	- Critério de informação de Akaike (do inglês <i>Akaike's Information Criterion</i> )
AM	- Modulação em ângulo (do inglês <i>Angle Modulation</i> )
AR	- Autorregressivo (do inglês <i>AutoRegressive</i> )
ARMA	- Autorregressivo com média móvel (do inglês <i>AutoRegressive Moving Average</i> )
ARMAX	- Autorregressivo com média móvel e com entrada externa (do inglês <i>AutoRegressive Moving Average with eXogenous input</i> )
ARX	- Autorregressivo com entrada externa (do inglês <i>AutoRegressive with eXogenous input</i> )
BA	- Algoritmo do Morcego (do inglês <i>Bat Algorithm</i> )
DE	- Evolução Diferencial (do inglês <i>Differential Evolution</i> )
ERR	- Taxa de redução do erro (do inglês <i>error reduction ratio</i> )
FIR	- Resposta ao impulso finita (FIR, do inglês <i>finite impulse response</i> )
GA	- Algoritmo Genético (do inglês <i>Genetic Algorithm</i> )
GVP	- Prioridade do maior valor (do inglês <i>Great Value Priority</i> )
MAM	- Modulação em ângulo modificada (do inglês <i>Modified Angle Modulation</i> )
MATLAB	- Matrix laboratory
MOSFET	- Transistor de efeito de campo de óxido de metal semicondutor (do inglês <i>Metal-Oxide-Semiconductor Field Effect Transistor</i> )
MQ	- Mínimos quadrados
NARMAX	- Não linear autorregressivo com média móvel e com entrada externa (do inglês <i>Nonlinear AutoRegressive Moving Average with eXogenous input</i> )
NARX	- Não linear autorregressivo com entrada externa (do inglês <i>Nonlinear AutoRegressive with eXogenous input</i> )
PRBS	- Sinal binário pseudoaleatório (do inglês <i>Pseudo Random Binary Signal</i> )
PWM	- Modulação por largura de pulso (do inglês <i>Pulse Width Modulation</i> )
RMSE	- Raiz quadrada do erro médio quadrático (do inglês <i>Root Mean Square Error</i> )
TF	- Função de transferência (do inglês <i>Transfer Function</i> )

## LISTA DE SÍMBOLOS

$a_{i,j,g_B}$	- $i$ -ésima componente do vetor $\mathbf{a}_{j,g_B}$ do BA
$\mathbf{a}_{j,g_B}$	- velocidade do $j$ -ésimo morcego na $g_B$ -ésima iteração do BA
$b_{i,j,g_B}$	- $i$ -ésima componente do vetor $\mathbf{b}_{j,g_B}$ do BA
$\mathbf{b}_{j,g_B}$	- posição do $j$ -ésimo morcego na $g_B$ -ésima iteração do BA
$c_j$	- custo do $j$ -ésimo cromossomo do GA
$C_j$	- custo normalizado do $j$ -ésimo cromossomo do GA
$Cr$	- probabilidade de cruzamento da DE
$d$	- atraso de transporte
$d_0$	- índice do vetor base da DE
$d_1, \dots, d_4$	- índices dos vetores diferença da DE
$D$	- razão cíclica do conversor <i>buck</i>
$e(k)$	- sinal de ruído no instante $k$
$f_{j,g_B}$	- frequência dos pulsos do $j$ -ésimo morcego na $g_B$ -ésima iteração do BA
$f_{max}$	- máxima frequência dos pulsos sonoros no BA
$f_{min}$	- mínima frequência dos pulsos sonoros no BA
$F^l\{\cdot\}$	- função com grau de não linearidade $l$
$F_m$	- fator de escala da DE
$g_B$	- contador de iterações do BA
$g_{B_{max}}$	- número máximo de iterações do BA
$g_D$	- contador de gerações da DE
$g_{D_{max}}$	- número máximo de gerações da DE
$g_G$	- contador de gerações do GA
$g_{G_{max}}$	- número máximo de gerações do GA
$g_j(\cdot)$	- $j$ -ésima função geradora da codificação AM
$h$	- parâmetro de escolha de bit na codificação TF
$h_B$	- fator de atualização da frequência no BA

$H_{j,g_B}$	- amplitude dos pulsos do $j$ -ésimo morcego na $g_B$ -ésima iteração do BA
$i, j$	- índices de vetores
$J(\cdot)$	- função objetivo
$J_B(\cdot)$	- função objetivo do BA
$J_D(\cdot)$	- função objetivo da DE
$J_G(\cdot)$	- função objetivo do GA
$J_{MQ}(\cdot)$	- função objetivo do MQ
$l$	- grau de não linearidade do modelo polinomial
$L$	- taxa de localização
$n$	- dimensão do problema de otimização
$n_e$	- máximo atraso do sinal de ruído
$n_t$	- soma dos máximos atrasos de entrada, de saída e de ruído (se houver)
$n_u$	- máximo atraso do sinal de entrada
$n_y$	- máximo atraso do sinal de saída
$n_\theta$	- número de termos em um modelo polinomial
$N$	- número de amostras em um conjunto de dados
$N_{P_B}$	- tamanho do grupo de morcegos no BA
$N_{P_D}$	- tamanho da população da DE
$N_{P_G}$	- tamanho da população do GA
$N_{sob}$	- número de cromossomos sobreviventes
$N_{V_B}$	- número de componentes em uma solução do BA
$N_{V_D}$	- número de componentes em uma solução da DE
$N_{V_G}$	- quantidade de genes em um cromossomo do GA
$N_\theta$	- tamanho do conjunto de termos candidatos
$p_i$	- $i$ -ésimo gene de um cromossomo
$p_m$	- probabilidade de mutação no GA
$P_j$	- probabilidade de reprodução do $j$ -ésimo cromossomo
$r_i$	- $i$ -ésimo regressor do conjunto de termos candidatos

$rand_i$	- número aleatório gerado na $i$ -ésima iteração
$RMSE$	- valor do RMSE da predição de um passo à frente
$RMSE_{SL}$	- valor do RMSE da simulação livre
$R^2$	- coeficiente de determinação da predição de um passo à frente
$R_{SL}^2$	- coeficiente de determinação da simulação livre
$R_\theta$	- conjunto de termos candidatos
$R^*$	- conjunto de termos selecionados
$s_1, s_2, s_3$	- indicadores da estratégia da DE
$S_t$	- total de simulações realizadas
$S_\alpha$	- quantidade de simulações em que o modelo de referência é localizado
$t_c$	- taxa de cruzamento do GA
$t_{sn}$	- taxa de seleção natural do GA
$T(\cdot)$	- função de transferência da codificação TF
$T_{ligado}$	- tempo em que a chave está ligada no conversor <i>buck</i>
$T_s$	- tempo de amostragem
$T_{s^*}$	- tempo de amostragem de trabalho
$T_t$	- período de operação da chave do conversor <i>buck</i>
$u(k)$	- sinal de entrada no instante $k$
$u_{i,j,g_D}$	- $i$ -ésima componente do vetor $\mathbf{u}_{j,g_D}$ da DE
$\mathbf{u}_{j,g_D}$	- $j$ -ésimo vetor tentativa na $g_D$ -ésima geração da DE
$v_d$	- tensão de entrada do conversor <i>buck</i>
$v_{i,j,g_D}$	- $i$ -ésima componente do vetor $\mathbf{v}_{j,g_D}$ da DE
$\mathbf{v}_{j,g_D}$	- $j$ -ésimo vetor mutante na $g_D$ -ésima geração da DE
$w, z$	- sinais genéricos
$\bar{w}, \bar{z}$	- valores médios dos sinais genéricos
$x_{i,j,g_D}$	- $i$ -ésima componente do vetor $\mathbf{x}_{j,g_D}$ da DE
$\mathbf{x}_{j,g_D}$	- $j$ -ésimo vetor alvo na $g_D$ -ésima geração da DE
$y(k)$	- sinal de saída no instante $k$

$\bar{y}$	- valor médio do sinal de saída
$\hat{y}$	- saída estimada através da predição de um passo à frente
$\hat{y}_{SL}$	- saída estimada através de simulação livre
$Y$	- vetor de saídas observadas
$\alpha_i$	- $i$ -ésima componente do vetor de soluções com valores contínuos
$A$	- vetor de soluções com valores contínuos
$\beta_i$	- $i$ -ésima componente do vetor de soluções com valores binários
$B$	- vetor de soluções com valores binários
$\gamma_i$	- $i$ -ésima componente do vetor $\Gamma$
$\Gamma$	- vetor de soluções intermediárias da codificação TF
$\delta(\tau)$	- impulso unitário
$\varepsilon$	- fator de busca local do BA
$\eta$	- constante de atualização de amplitudes de pulsos do BA
$\vartheta$	- constante de atualização de taxa de emissão de pulsos do BA
$\theta_i$	- $i$ -ésimo parâmetro do modelo polinomial
$\hat{\theta}_i$	- $i$ -ésimo parâmetro estimado
$\hat{\Theta}$	- vetor de parâmetros estimados
$\lambda_i$	- $i$ -ésima componente do vetor $\Lambda$
$\Lambda$	- vetor de posições da codificação GVP
$\mu$	- fator de forma da função $T(\cdot)$
$\xi(k)$	- resíduo no instante $k$
$\Xi$	- vetor de resíduos
$\rho_{ij}$	- $i$ -ésimo elemento do vetor $\boldsymbol{\rho}_j$
$\boldsymbol{\rho}_j$	- vetor de coeficientes da $j$ -ésima função geradora da codificação AM
$\sigma_{erro}^2$	- variância dos resíduos
$\phi_{wz}(\tau)$	- correlação cruzada entre os sinais $w$ e $z$
$\varphi(k)$	- vetor de regressores no instante $k$



$\Psi$  - matriz de regressores

$\omega_{j,g_B}$  - taxa de emissão de pulsos do  $j$ -ésimo morcego na  $g_B$ -ésima iteração do BA

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>16</b>
1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO DA PESQUISA .....	18
1.2 OBJETIVOS .....	20
1.2.1 Objetivo geral .....	20
1.2.2 Objetivos específicos.....	20
1.3 ORGANIZAÇÃO DO TRABALHO .....	21
<b>2 IDENTIFICAÇÃO DE SISTEMAS .....</b>	<b>22</b>
2.1 INTRODUÇÃO .....	22
2.2 COLETA DE DADOS EXPERIMENTAIS .....	23
2.3 REPRESENTAÇÕES MATEMÁTICAS PARA SISTEMAS .....	24
2.3.1 Modelos polinomiais NARMAX.....	25
2.4 SELEÇÃO DE ESTRUTURA.....	28
2.5 ESTIMAÇÃO DE PARÂMETROS .....	30
2.5.1 Método dos Mínimos Quadrados .....	30
2.6 VALIDAÇÃO DE MODELOS .....	32
2.6.1 Simulação.....	33
2.6.1.1 Predição de um passo à frente.....	33
2.6.1.2 Simulação livre .....	34
2.6.1.3 Medidas de desempenho .....	35
2.6.2 Validação estatística .....	36
2.7 RESUMO DO CAPÍTULO .....	38
<b>3 METAHEURÍSTICAS .....</b>	<b>39</b>
3.1 INTRODUÇÃO .....	39
3.2 ALGORITMO GENÉTICO .....	39
3.2.1 Seleção natural .....	41
3.2.2 Seleção para reprodução .....	42
3.2.2.1 Seleção por emparelhamento do melhor ao pior.....	42
3.2.2.2 Seleção por roleta.....	42
3.2.2.3 Seleção por torneio.....	43
3.2.3 Cruzamento.....	43
3.2.4 Mutação.....	45
3.3 EVOLUÇÃO DIFERENCIAL.....	46

3.3.1 Mutação.....	47
3.3.2 Cruzamento .....	48
3.3.3 Seleção .....	49
3.3.4 Estratégias .....	49
3.4 ALGORITMO DO MORCEGO.....	50
3.4.1 Movimentação dos morcegos.....	51
3.4.2 Amplitude de pulsos e taxa de emissão dos pulsos .....	52
3.5 RESUMO DO CAPÍTULO .....	53
<b>4 CODIFICAÇÃO BINÁRIA PARA METAHEURÍSTICAS.....</b>	<b>54</b>
4.1 INTRODUÇÃO .....	54
4.2 CODIFICAÇÃO BINÁRIA POR FUNÇÃO DE TRANSFERÊNCIA .....	55
4.3 CODIFICAÇÃO BINÁRIA POR PRIORIDADE DO MAIOR VALOR .....	56
4.4 CODIFICAÇÃO BINÁRIA POR MODULAÇÃO EM ÂNGULO .....	56
4.5 CODIFICAÇÃO BINÁRIA POR MODULAÇÃO EM ÂNGULO MODIFICADA.....	58
4.6 RESUMO DO CAPÍTULO .....	58
<b>5 METODOLOGIA .....</b>	<b>60</b>
5.1 INTRODUÇÃO .....	60
5.2 OBTENÇÃO DOS DADOS E SEPARAÇÃO EM DOIS CONJUNTOS .....	60
5.3 CRIAÇÃO DO CONJUNTO DE TERMOS CANDIDATOS .....	60
5.4 ESCOLHA DA METAHEURÍSTICA.....	62
5.5 POPULAÇÃO INICIAL E AVALIAÇÃO DA APTIDÃO DE SOLUÇÕES .....	63
5.6 EVOLUÇÃO DA POPULAÇÃO DE SOLUÇÕES .....	64
5.7 AVALIAÇÃO DO DESEMPENHO DAS TÉCNICAS .....	64
5.8 RESUMO DO CAPÍTULO .....	67
<b>6 ESTUDOS DE CASO.....</b>	<b>68</b>
6.1 INTRODUÇÃO .....	68
6.2 ESTUDO DE CASO 1: CONVERSOR <i>BUCK</i> .....	68
6.2.1 Parâmetros do algoritmo .....	70
6.2.2 Desempenho das técnicas .....	73
6.2.3 Validação dos modelos obtidos.....	83
6.2.4 Comparações com a literatura .....	90
6.3 ESTUDO DE CASO 2: AQUECEDOR ELÉTRICO .....	94
6.3.1 Parâmetros do algoritmo .....	98
6.3.2 Desempenho das técnicas .....	99

6.3.3 Validação dos modelos obtidos.....	111
6.3.4 Comparações com a literatura .....	120
<b>7 CONSIDERAÇÕES FINAIS .....</b>	<b>123</b>
7.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS .....	125
<b>REFERÊNCIAS.....</b>	<b>126</b>

## 1 INTRODUÇÃO

A engenharia é a área dedicada ao entendimento e controle de materiais e forças da natureza para o benefício da humanidade. Sua principal tarefa é compreender e controlar segmentos do ambiente, geralmente chamados de sistemas, visando o fornecimento de produtos úteis à sociedade. Dentro deste contexto, um sistema pode significar, por exemplo, um processo químico, um manipulador robótico, uma linha de produção industrial, etc.

Uma maneira de compreender um sistema é através de sua representação por meio de modelos matemáticos, o que sempre se mostrou uma tarefa desafiadora. O homem sempre vem buscando maneiras de descrever matematicamente sistemas experimentais para facilitar a tarefa de entendê-los e, assim, resolver problemas relacionados a eles (AGUIRRE, 2015).

Existem diversas razões que levam à necessidade de se obter um modelo para um sistema real. Em primeiro lugar, testes com sistemas reais podem sair caro e, em alguns casos, podem até ser perigosos. Uma outra razão possível é a inexistência do sistema real na configuração analisada. Vale ressaltar que essa busca pelo modelo do sistema real geralmente é motivada por aplicações práticas como simulação, previsão, otimização ou projeto de sistemas de controle e pode ocorrer em diversas áreas tais como ciências naturais, especialmente física, química e biologia, e também nas áreas de engenharia médica e economia (ISERMANN; MÜNCHHOF, 2011).

Se as leis físicas de um determinado sistema são conhecidas na forma analítica, seu comportamento básico, estático ou dinâmico, pode ser obtido por modelagem fenomenológica ou modelagem baseada na física do processo. Entretanto, se estas leis são desconhecidas, ou conhecidas apenas parcialmente, deve-se realizar uma modelagem experimental ou empírica, que é denominada de identificação de sistemas (AGUIRRE, 2015; ISERMANN; MÜNCHHOF, 2011).

Identificação de sistemas é a técnica utilizada para obter modelos de um sistema a partir de medições das suas entradas e saídas (BILLINGS, 2013). O campo científico da identificação de sistemas foi sistematicamente desenvolvido desde os anos 1960, especialmente nas áreas de controle e engenharia de comunicação. É baseado nos métodos da teoria dos sistemas, teoria dos sinais, teoria do controle e teoria da estimativa estatística e foi influenciado pelas modernas técnicas de medição,



cálculos digitais e pela necessidade de funções precisas para processamento e controle de sinais (ISERMANN; MÜNCHHOF, 2011).

O procedimento de identificação de sistemas é composto pelas seguintes etapas básicas: testes dinâmicos e coleta de dados; escolha da representação matemática a ser usada; determinação da estrutura do modelo; estimação dos parâmetros; e validação do modelo (AGUIRRE, 2015).

A coleta de dados é a etapa em que serão realizadas medidas das entradas e saídas do sistema, com a finalidade de gerar os dados necessários ao procedimento de identificação (YASSIN; TAIB; ADNAN, 2013). Como a qualidade dos dados irá interferir nos modelos obtidos, deve-se atentar para que o sistema seja excitado com os sinais adequados. Diversos trabalhos na literatura apresentam estudos específicos para esta etapa (ABRASHOV et al., 2018; HUNG et al., 2015; VAILLANT; KURAMOTO; GARCIA, 2013).

A escolha da representação matemática consiste em determinar, basicamente, o tipo de modelo que será utilizado. Caso decida-se por uma representação linear, existe a família de modelos autorregressivos com média móvel e com entradas externas (ARMAX, do inglês *AutoRegressive Moving Average with eXogenous input*) (BILLINGS, 2013). Por outro lado, ao se optar por modelos não lineares, é possível utilizar a classe de modelos não lineares autorregressivos com média móvel e com entradas externas (NARMAX, do inglês *Nonlinear AutoRegressive Moving Average with eXogenous input*) (BILLINGS; VOON, 1983; LEONTARITIS; BILLINGS, 1985a, 1985b), além de redes neurais artificiais (LIU, 2001; NARENDRA; PARTHASARATHY, 1990) e sistemas *fuzzy* (NELLES, 2000).

A determinação da estrutura do modelo consiste em detectar, dentro de um conjunto de termos possíveis, quais destes termos serão incluídos no modelo (AGUIRRE; RODRIGUES; JÁCOME, 1998). Há uma série de abordagens para atingir este propósito, dentre as quais destacam-se: abordagens baseadas na taxa de redução do erro (ERR, do inglês *error reduction ratio*) (BILLINGS; CHEN; KORENBERG, 1989; KORENBERG et al., 1988) ou no critério de informação de Akaike (AKAIKE, 1974); abordagens que aplicam metaheurísticas (FONSECA et al., 1993); e abordagens de seleção randômica (FALSONE; PIRODDI; PRANDINI, 2015).

Com a estrutura do modelo determinada, o próximo passo é realizar a estimação de seus parâmetros. Nesta etapa, é comum a aplicação do método dos mínimos quadrados (MQ) clássico ou alguma de suas variantes, como por exemplo

MQ estendido, MQ ponderado, MQ recursivo, MQ generalizado (AGUIRRE, 2015; COELHO; COELHO, 2015), ou ainda o MQ ortogonal, que combina seleção de estrutura e estimação de parâmetros (KORENBERG et al., 1988).

A última etapa consiste em validar os modelos obtidos, ou seja, verificar se os mesmos incorporam as características de interesse do sistema original. As principais maneiras de realizar esta verificação são simulações e análise de resíduos. Além disso, é interessante poder comparar modelos entre si e decidir se há algum candidato significativamente melhor que os demais (AGUIRRE, 2015).

## 1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO DA PESQUISA

Dentro da identificação de sistemas, a etapa de seleção da estrutura constitui uma difícil tarefa. A escolha dos termos apropriados para compor um modelo geralmente é um processo complexo, especialmente no caso de modelos polinomiais, onde a quantidade de combinações possíveis pode atingir um número excessivamente elevado.

Muitos trabalhos vêm sendo desenvolvidos com o intuito de determinar quais termos são essenciais para o modelo de um sistema. Algumas abordagens classificam os termos de acordo com algum grau de contribuição para o modelo, usando como base, por exemplo, a taxa de redução do erro (ERR, do inglês *error reduction ratio*) (BILLINGS; CHEN; KORENBERG, 1989; KORENBERG et al., 1988). Por outro lado, há técnicas que fazem essa busca pelos melhores termos de uma maneira randômica (BIANCHI et al., 2017; FALSONE; PIRODDI; PRANDINI, 2015; RETES; AGUIRRE, 2019).

A busca pelos termos mais adequados para um modelo pode ser tratada como um problema de otimização, abrindo espaço para abordagens que aplicam metaheurísticas neste processo. Metaheurísticas são técnicas aplicadas em problemas complexos de otimização e, em sua grande maioria, imitam metáforas naturais para atingir este propósito (BOUSSAÏD; LEPAGNOT; SIARRY, 2013; DU; SWAMY, 2016; SÖRENSEN; GLOVER, 2013). Alguns exemplos de sua aplicação em problemas de seleção de estrutura são: Algoritmo Genético (FONSECA et al., 1993), Programação Genética (MADÁR; ABONYI; SZEIFERT, 2005) e Otimização por Enxame de Partículas (HAFIZ et al., 2018).

O Algoritmo Genético realiza busca e otimização com base nos princípios da genética e da seleção natural (HAUPT; HAUPT, 2004), e têm se mostrado uma boa escolha para realizar a seleção de estrutura dentro do contexto de identificação de sistemas. Fonseca et al. (1993) apresentaram um dos primeiros estudos que utilizaram Algoritmo Genético para localizar o conjunto de termos de um modelo e seus respectivos coeficientes. Entretanto, neste trabalho, o número de termos do modelo não era levado em consideração, o que foi analisado posteriormente em uma formulação multiobjetiva do Algoritmo Genético (FONSECA; FLEMING, 1996).

Luh e Wu (1999) avaliaram os efeitos de diferentes combinações de operadores do Algoritmo Genético aplicados na seleção de estrutura para modelos de um sistema de nível de líquido acoplado. No trabalho de Kumon et al. (2000), modelos polinomiais são obtidos para um sistema de vibração ressonante de duas massas.

Ahmad, Jamaluddin e Hussain (2004) aplicaram uma versão modificada do Algoritmo Genético na obtenção de modelos ARMAX e NARMAX de alguns sistemas teóricos, estabelecendo ainda um comparativo entre os desempenhos das versões básica e modificada, bem como dos mínimos quadrados ortogonais.

Além do seu potencial em solucionar problemas de otimização, a popularidade do Algoritmo Genético neste tipo de aplicação advém também do fato destes utilizarem uma codificação binária na manipulação das soluções, o que facilita a representação dos possíveis termos para um modelo polinomial. Embora haja outras codificações possíveis para o Algoritmo Genético, a binária mostrou destaque em problemas de seleção de estrutura para modelos polinomiais, quando comparada a outras codificações (HO et al., 1998).

Desta forma, algumas metaheurísticas projetadas para atuar num espaço de busca contínuo vêm sendo adaptadas para manipular soluções binárias em diversas aplicações, dentre elas, o problema de seleção de estrutura. Como exemplos, podem ser citados Otimização por Enxame de Partículas (HAFIZ et al., 2018; HAFIZ; SWAIN; MENDES, 2019; KENNEDY; EBERHART, 1995), Evolução Diferencial (OLIVEIRA; LEANDRO, 2019; PRICE, 1996; ZAKARIA et al., 2014, 2015, 2018) e Algoritmo do Morcego (SEVERINO; ARAÚJO, 2017; YANG, 2010).

Entretanto, a literatura aponta que metaheurística originalmente criadas para espaços contínuos podem ter suas versões binárias implementadas de várias maneiras (CRAWFORD et al., 2017), o que sugere um interessante campo de investigação associado ao problema de seleção de estrutura.

Dada a diversidade de métodos de codificação binária, esta pesquisa tem por finalidade investigar o impacto causado pela escolha da codificação binária em dois passos na seleção de estrutura de modelos polinomiais, comparando os resultados obtidos com aqueles alcançados por técnicas já consagradas neste tipo de problema, como por exemplo, o Algoritmo Genético.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

O objetivo geral deste trabalho é avaliar o impacto dos tipos de codificação binária associados a metaheurísticas geradas para espaços de busca contínuos aplicadas na seleção de estrutura de modelos polinomiais para fins de identificação de sistemas.

### 1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Aplicar metaheurísticas originalmente binárias na seleção de estrutura de modelos polinomiais NARX para sistemas previamente escolhidos;
- Realizar seleção de estrutura de modelos polinomiais NARX com metaheurísticas projetadas para espaços contínuos associadas a codificações binárias;
- Verificar a influência do tipo de codificação binária no desempenho de metaheurísticas originalmente projetadas para espaços de busca contínuos aplicadas na seleção de estrutura;
- Comparar o desempenho de versões binárias de metaheurísticas voltadas para o espaço de busca contínuo com o de metaheurísticas voltadas para o espaço de busca binário;
- Avaliar os modelos obtidos aplicando métodos de validação;
- Comparar os modelos obtidos para os sistemas escolhidos com modelos de referência da literatura.

### 1.3 ORGANIZAÇÃO DO TRABALHO

O restante deste trabalho está organizado conforme as explicações a seguir. No capítulo 2, são abordados os conceitos básicos de identificação de sistemas, necessários para o desenvolvimento desta dissertação. O capítulo 3 contextualiza o leitor sobre as técnicas conhecidas como metaheurísticas e aborda de maneira um pouco mais detalhada aquelas que serão aplicadas neste trabalho. No capítulo 4 são apresentadas as técnicas de codificação binária que serão associadas com algumas metaheurísticas originalmente projetadas para atuar num espaço de busca contínuo. A metodologia utilizada no decorrer deste trabalho é apresentada no capítulo 5. O capítulo 6 apresenta os estudos de caso em que foram aplicadas as técnicas apresentadas nos capítulos anteriores. Por fim, no capítulo 7, são apresentadas as considerações finais e apresentados os caminhos futuros desta pesquisa.

## 2 IDENTIFICAÇÃO DE SISTEMAS

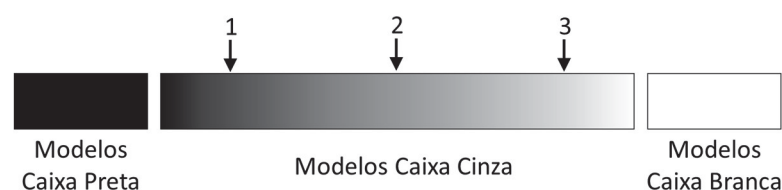
### 2.1 INTRODUÇÃO

Identificação de sistemas pode ser definida como um conjunto de procedimentos que se propõem a obter um modelo matemático que explique, pelo menos em parte e de forma aproximada, a relação de causa e efeito presente nos dados provenientes de um sistema (AGUIRRE, 2015).

À medida que informações adicionais são incorporadas ao procedimento de identificação, este tende a transitar entre os extremos da modelagem fenomenológica (baseada na física do processo) e da modelagem empírica (identificação de sistemas). A depender do número de informações disponíveis sobre o sistema a ser analisado, os métodos de identificação podem ser classificados em três tipos: caixa-preta, onde nenhuma informação sobre o sistema está disponível além dos dados de entrada e saída; caixa-branca ou simplesmente modelagem física, no qual todos os parâmetros são conhecidos, ou previamente determinados; e caixa-cinza, que busca combinar as vantagens dos procedimentos de identificação caixa-preta e caixa-branca. Nesse caso, tanto dados de entrada e saída obtidos no sistema, quanto informação auxiliar são usados na identificação (CORRÊA; AGUIRRE, 2004).

Na FIGURA 1 é possível observar que a tonalidade do “cinza” varia de acordo com o nível e a quantidade de informação auxiliar usada no processo de modelagem. Procedimentos em que o uso de informação auxiliar se limita à definição de termos candidatos, no caso de um modelo polinomial, se situariam na região 1. Já procedimentos em que se usa informação auxiliar na estimação de parâmetros se situam na região 2. Por fim, os casos em que a estrutura é definida por leis físicas e parâmetros estimados com dados de entrada e saída são os procedimentos situados na região 3 (CORRÊA; AGUIRRE, 2004).

FIGURA 1 – Escala com diferentes níveis de informação auxiliar na identificação caixa cinza



FONTE: Adaptado de Corrêa e Aguirre (2004).

Em linhas gerais, as principais etapas de um problema de identificação são (AGUIRRE, 2015):

- a) Coleta de dados experimentais: para que um processo de identificação de um sistema possa ocorrer é necessário um conjunto de dados que relacione a entrada e a saída do sistema;
- b) Escolha da representação matemática: deve-se determinar qual o formato do modelo que será gerado a partir dos dados do sistema;
- c) Determinação da estrutura do modelo: nesta etapa escolhe-se a estrutura do modelo matemático que será usada pra representar o sistema, através de métodos de detecção de estrutura;
- d) Estimação dos parâmetros do modelo: nesta etapa os valores numéricos que compõem o modelo matemático são estimados através dos diversos métodos de estimação de parâmetros;
- e) Validação do modelo: tendo obtido um o modelo candidato, é necessário verificar se ele representa as características de interesse do sistema real.

As próximas seções descreverão os principais aspectos que compõem cada uma das etapas supracitadas.

## 2.2 COLETA DE DADOS EXPERIMENTAIS

A coleta de dados é a tarefa de obtenção de dados de entrada e saída de um sistema para fins de identificação. É uma etapa importante, pois o procedimento de identificação depende muito da qualidade dos dados (YASSIN; TAIB; ADNAN, 2013). Nesta etapa, o sistema deve ser excitado através da aplicação de entradas pré-definidas, as quais devem apresentar espectro suficientemente amplo em frequência e amplitude de modo a excursionar o sistema pelos regimes dinâmicos de interesse. Um tipo de sinal bastante utilizado como entrada para identificação de sistemas são os sinais binários pseudoaleatórios (PRBS, do inglês *Pseudo Random Binary Signal*), embora outros tipos também sejam aplicáveis, como por exemplo ondas quadradas, senoides e ruídos brancos ou gaussianos (AGUIRRE, 2015; AGUIRRE; RODRIGUES; JÁCOME, 1998; YASSIN; TAIB; ADNAN, 2013).

Outra característica importante dos sinais usados para identificação é o seu período de amostragem,  $T_s$ , definido como o intervalo de tempo entre duas amostras. Sua escolha deve ser feita com cautela, pois afetará diretamente a determinação da estrutura do modelo. Um período de amostragem grande demais pode fazer com que determinadas interações não lineares não apareçam e algumas frequências de interesse não sejam visitadas. Por outro lado, se o período de amostragem for muito menor que o necessário, haverá redundância entre amostras vizinhas (AGUIRRE; BILLINGS, 1995). Uma maneira de determinar o período de amostragem seria através do teorema de Shannon, o qual diz que um sinal que não contenha componentes de frequência acima de  $1/2T_s$  pode ser determinado unicamente a partir de amostras de tal sinal separadas por  $T_s$ . Na prática, entretanto, a frequência de amostragem é normalmente escolhida entre 5 e 10 vezes maior do que a maior frequência de interesse contida nos dados, e não apenas 2 vezes maior (chamada de frequência de Nyquist), como exigido pelo teorema de Shannon (AGUIRRE, 2015).

Como será visto nas seções posteriores, recomenda-se que o conjunto de dados gerados (entradas e saídas) seja dividido em dois conjuntos distintos: o primeiro irá conter os dados de estimação, utilizados para construir o modelo do sistema; o segundo será composto pelos dados de validação, utilizados para avaliar o desempenho do modelo. Desta forma, os modelos obtidos no conjunto de dados de estimação são, portanto, testados em um conjunto de dados independente que não foi usado no processo de estimativa dos modelos (BILLINGS, 2013).

Neste trabalho, serão utilizados dados coletados através de experimentos realizados por outros pesquisadores e que encontram-se disponíveis na literatura.

## 2.3 REPRESENTAÇÕES MATEMÁTICAS PARA SISTEMAS

Na modelagem de sistemas, uma importante questão é a escolha do tipo de representação matemática que será utilizada. Esta representação deve ser genérica o suficiente para incluir os aspectos relevantes do sistema (AGUIRRE, 2019).

Uma primeira decisão a ser tomada é se o modelo será linear ou não linear. Um tipo de representação linear muito comum são os modelos do tipo autorregressivo com média móvel e com entradas externas (ARMAX, do inglês *AutoRegressive Moving Average with eXogenous input*). Neste grupo encontram-se os casos



particulares: modelo de resposta ao impulso finita (FIR, do inglês *Finite Impulse Response*) modelo autorregressivo (AR, do inglês *AutoRegressive*), modelo autorregressivo com entradas externas (ARX, do inglês *AutoRegressive with eXogenous input*), modelo autorregressivo com média móvel (ARMA, do inglês *AutoRegressive Moving Average*), modelo de erro na saída (OE, do inglês *Output-Error*) e modelo Box-Jenkins (BJ) (BILLINGS, 2013).

Dentre as representações não lineares, podem ser citadas as redes neurais artificiais (HAYKIN, 2001; LIU, 2001; NARENDRA; PARTHASARATHY, 1990; PHAM; LIU, 1995), séries de Volterra (SCHETZEN, 1980) e sistemas *fuzzy* (NELLES, 2000). Além destas representações não lineares, existe ainda a classe de modelos não lineares autorregressivos com média móvel e com entradas externas (NARMAX, do inglês *Nonlinear AutoRegressive Moving Average with eXogenous input*) (BILLINGS; VOON, 1983; LEONTARITIS; BILLINGS, 1985a, 1985b), que constituem as representações mais populares para identificação de sistemas dinâmicos não lineares no domínio do tempo discreto. Apesar de ser uma representação não linear, este tipo de modelo é linear nos parâmetros, o que o torna particularmente conveniente, pois a estimação de seus parâmetros pode ser feita com estimadores lineares, como o método dos mínimos quadrados. Neste trabalho será aplicado um subtipo do modelo NARMAX, conhecido como modelo não linear autorregressivo com entradas externas (NARX, do inglês *Nonlinear AutoRegressive with eXogenous input*). Os aspectos matemáticos de ambos serão abordados a seguir.

### 2.3.1 Modelos polinomiais NARMAX

Os modelos NARMAX são representações que explicam o valor da saída de um sistema em função de valores prévios dos sinais de saída, de entrada e de ruído. O modelo NARMAX é definido como:

$$\begin{aligned} y(k) = F^l\{ & y(k-1), y(k-2), \dots, y(k-n_y), \\ & u(k-d), u(k-d-1), \dots, u(k-d-n_u), \\ & e(k-1), e(k-2), \dots, e(k-n_e)\} + e(k) \end{aligned} \quad (2.1)$$

onde  $y(k)$ ,  $u(k)$  e  $e(k)$  são, respectivamente, a saída, a entrada e o ruído do sistema;  $n_y$ ,  $n_u$  e  $n_e$  são os máximos atrasos da saída, da entrada e do ruído, respectivamente;  $F^l\{\cdot\}$  é alguma função com grau de não linearidade  $l \in \mathbb{N}$ ; os termos dentro da função  $F^l\{\cdot\}$  são os regressores; e  $d$  é o atraso de transporte. De modo geral, o sinal  $e(k)$  representa os erros de modelagem, o ruído aditivo do sistema e incertezas de ordem qualquer, sendo também chamado de resíduo. É desejável que os resíduos sejam aleatórios e não contenham informação útil acerca dos dados, ou seja, espera-se que tenham o comportamento de um ruído branco.

A forma da função  $F^l\{\cdot\}$  geralmente é desconhecida, sendo necessário reconstruir a dinâmica do sistema utilizando uma aproximação desta função. As aproximações mais utilizadas são a polinomial e a racional (CHEN; BILLINGS, 1989). A representação polinomial do modelo NARMAX é dada por:

$$y(k) = \theta_0 + \sum_{i_1=1}^{n_t} f_{i_1}(q_{i_1}(k)) + \sum_{i_1=1}^{n_t} \sum_{i_2=i_1}^{n_t} f_{i_1 i_2}(q_{i_1}(k), q_{i_2}(k)) + \dots$$

$$+ \sum_{i_1=1}^{n_t} \dots \sum_{i_l=i_{l-1}}^{n_t} f_{i_1 i_2 \dots i_l}(q_{i_1}(k), q_{i_2}(k), \dots, q_{i_l}(k)) + e(k) \quad (2.2)$$

Para esta representação, devem ser realizadas as seguintes considerações:

$$f_{i_1 i_2 \dots i_m}(q_{i_1}(k), q_{i_2}(k), \dots, q_{i_m}(k)) = \theta_{i_1 i_2 \dots i_m} \prod_{k=1}^m q_{i_k}(k), 1 \leq m \leq l \quad (2.3)$$

$$q_m(k) = \begin{cases} y(k-m), & 1 \leq m \leq n_y \\ u(k-(m-n_y)), & n_y + 1 \leq m \leq n_y + n_u \\ e(k-(m-n_y-n_u)), & n_y + n_u + 1 \leq m \leq n_y + n_u + n_e \end{cases} \quad (2.4)$$

onde  $l$  é o grau de não linearidade do polinômio,  $\theta_{i_1 i_2 \dots i_m}$  são os parâmetros do modelo e  $n_t = n_y + n_u + n_e$ .

Mais especificamente, a equação (2.2) pode ser explicitamente escrita como:

$$\begin{aligned}
 y(k) = & \theta_0 + \sum_{i_1=1}^{n_t} \theta_{i_1} q_{i_1}(k) + \sum_{i_1=1}^{n_t} \sum_{i_2=i_1}^{n_t} \theta_{i_1 i_2} q_{i_1}(k) q_{i_2}(k) + \dots \\
 & + \sum_{i_1=1}^{n_t} \dots \sum_{i_l=i_{l-1}}^{n_t} \theta_{i_1 i_2 \dots i_l} q_{i_1}(k) q_{i_2}(k) \dots q_{i_l}(k) + e(k)
 \end{aligned} \tag{2.5}$$

Um caso especial do modelo NARMAX é o modelo NARX, que não inclui nenhum termo dependente do ruído, como por exemplo  $y(k-1)e(k-2)$ , etc. Desta forma, a definição de  $q_m(k)$  na equação (2.4) passa a ser:

$$q_m(k) = \begin{cases} y(k-m), & 1 \leq m \leq n_y \\ u(k-(m-n_y)), & n_y + 1 \leq m \leq n_t = n_y + n_u \end{cases} \tag{2.6}$$

Com esta definição, o modelo NARX pode ser implicitamente formulado como:

$$\begin{aligned}
 y(k) = & F^l \{ y(k-1), y(k-2), \dots, y(k-n_y), \\
 & u(k-d), u(k-d-1), \dots, u(k-d-n_u) \} + e(k)
 \end{aligned} \tag{2.7}$$

Considerando, por exemplo,  $d = 1$ ,  $l = 2$ ,  $n_y = 1$  e  $n_u = 1$ , o modelo NARX obtido seria:

$$\begin{aligned}
 y(k) = & \theta_0 + \theta_1 y(k-1) + \theta_2 u(k-1) + \theta_{11} y^2(k-1) + \\
 & \theta_{12} y(k-1) u(k-1) + \theta_{22} u^2(k-1) + e(k)
 \end{aligned} \tag{2.8}$$

Como os modelos do processo e do ruído podem, na prática, ser desacoplados, é possível iniciar a modelagem utilizando um modelo NARX e

posteriormente ir adicionando termos de ruído, caso a validação do modelo indique que os resíduos não são brancos (BILLINGS; CHEN; KORENBERG, 1989).

## 2.4 SELEÇÃO DE ESTRUTURA

A seleção da estrutura para um modelo é uma tarefa essencial e configura-se como a mais árdua no processo de identificação de sistemas não lineares. Em um modelo polinomial, a união de todos os termos que podem compor a sua estrutura denomina-se conjunto de termos candidatos. Desta forma, a seleção de estrutura consiste em detectar quais termos serão incluídos no modelo (AGUIRRE; RODRIGUES; JÁCOME, 1998).

O número de termos que compõem o conjunto de termos candidatos para modelos NARMAX é dado por:

$$N_{\theta} = \frac{(n + l)!}{n! l!} \quad (2.9)$$

onde  $l$  é o grau de não linearidade e  $n = n_y + n_u + n_e$ . Para o caso do modelo NARX, basta assumir  $n = n_y + n_u$ . Observa-se que, para valores grandes de  $l$ ,  $n_y$ ,  $n_u$  e  $n_e$ , a quantidade de termos candidatos pode ser excessiva. Entretanto, na maioria dos casos práticos, somente uma pequena quantidade de termos são importantes para descrever a dinâmica do sistema e, com isso, o desafio consiste em determinar quais são estes termos (BILLINGS, 2013).

O problema de seleção de estrutura pode ser definido da seguinte forma: dado um conjunto de termos candidatos  $R_{\theta} = \{r_1, r_2, \dots, r_{N_{\theta}}\}$ , onde o  $i$ -ésimo termo  $r_i$  pode representar qualquer um dos regressores em um modelo polinomial, o objetivo da seleção de estrutura é determinar um subconjunto de termos significantes,  $R^* \subset R_{\theta}$ , que otimizam alguma função  $J(\cdot)$  adequada, conforme:

$$R^* = \arg \min_{R \subset R_{\theta}} J(R) \quad (2.9)$$

Trata-se de um problema de otimização combinatória e a busca exaustiva envolveria a avaliação de  $2^{N_\theta} - 1$  subconjuntos, o que geralmente é algo impraticável (HAFIZ; SWAIN; MENDES, 2019).

As técnicas para seleção de estrutura de um modelo não linear podem ser agrupadas em duas categorias, denominadas construtiva e eliminativa. Em algumas técnicas a estrutura inicial do modelo é um conjunto vazio e a cada passo seguinte o termo mais importante entre todos os termos candidatos é anexado ao modelo. Tais técnicas podem ser chamadas de construtivas, pois constroem a estrutura do modelo gradualmente a cada passo (AGUIRRE, 1994). Um exemplo deste tipo de procedimento é o critério conhecido como taxa de redução do erro (ERR, do inglês *error reduction ratio*). O ERR de cada termo candidato é um número que indica a melhoria obtida na representação do sistema através da sua inclusão no modelo. Desta forma, a cada passo o termo com o maior ERR é incluído na estrutura do modelo e, após um certo número de termos serem incluídos, os parâmetros são estimados (BILLINGS; CHEN; KORENBERG, 1989; KORENBERG et al., 1988).

Por outro lado, existem abordagens em que um modelo experimental é construído e, subsequentemente, os termos menos importantes do modelo são excluídos. Nestas técnicas, chamadas de eliminativas, às vezes é necessário realizar a estimação de parâmetros para o modelo de teste, que pode ter um número bastante elevado de parâmetros (AGUIRRE, 1994).

Outro tipo de abordagem empregada na seleção de estrutura são os critérios de informação, sendo o critério de informação de Akaike (AIC, do inglês *Akaike's Information Criterion*) um dos mais populares. O AIC é um critério estatístico que verifica a redução na variância dos resíduos, à medida que termos são incorporados ao modelo (AKAIKE, 1974). Este critério pode ser empregado para determinar o número de termos no modelo e é dado por:

$$AIC(n_\theta) = N \ln[\sigma_{erro}^2(n_\theta)] + 2n_\theta \quad (2.10)$$

sendo que  $N$  é o número de amostras,  $\sigma_{erro}^2(n_\theta)$  é a variância dos resíduos e  $n_\theta$  é o número de termos que compõem o modelo.

Diversas outras abordagens podem ser empregadas na etapa de seleção de estrutura, como por exemplo, análise modal (THOUVEREZ; JEZEQUEL, 1996),

estatística baseada nos erros de predição (WANG; CLUETT, 1996), métodos de clusterização (JUANG; HSIEH, 2010; TALMON et al., 2011) e abordagens randômicas (FALSONE; PIRODDI; PRANDINI, 2015; RETES; AGUIRRE, 2019).

Existem ainda abordagens que aplicam metaheurísticas na busca pelos termos mais adequados para um modelo (FONSECA et al., 1993; HAFIZ et al., 2018; MADÁR; ABONYI; SZEIFERT, 2005; ZAKARIA et al., 2018). Neste tipo de abordagem, a qual será aplicada neste trabalho, as metaheurísticas manipulam soluções em formato binário dentro de um espaço de busca, visando minimizar alguma função, geralmente associada a alguma medida de erro ou de complexidade do modelo. Considerando o conjunto de termos candidatos  $R_\theta = \{r_1, r_2, \dots, r_{N_\theta}\}$  e a solução  $B = \{\beta_1, \beta_2, \dots, \beta_{N_\theta}\}$ ,  $\beta_i \in \{0,1\}$ , o  $i$ -ésimo termo de  $R_\theta$  será incluído no modelo se o valor de  $\beta_i$  for igual a 1 (HAFIZ; SWAIN; MENDES, 2019). Desta forma a quantidade  $n_\theta$  de termos que irão compor o modelo é igual à quantidade de elementos iguais a 1 em  $B$ .

## 2.5 ESTIMAÇÃO DE PARÂMETROS

Uma vez escolhida a classe de modelos a ser utilizada e selecionada uma estrutura dentro desta classe, o próximo passo para obter o modelo é a estimação de seus parâmetros. Nesta etapa determinam-se os valores numéricos dos parâmetros com o objetivo de aproximar o comportamento dinâmico apresentado pelo sistema original (AGUIRRE; RODRIGUES; JÁCOME, 1998). Neste ponto é necessário determinar qual algoritmo será aplicado para realizar a estimação dos parâmetros. A maior parte dos algoritmos empregados com esta finalidade são variações do método clássico dos mínimos quadrados (AGUIRRE, 2015), o qual será aplicado neste trabalho.

### 2.5.1 Método dos Mínimos Quadrados

O método dos Mínimos Quadrados (MQ) é um dos algoritmos mais conhecidos e mais utilizados nas mais diversas áreas de ciência e tecnologia (AGUIRRE, 2015). Foi desenvolvido por Karl Friedrich Gauss em 1795 para prever a

trajetória de planetas e cometas a partir de observações realizadas. O estimador MQ busca o melhor ajuste para um conjunto de dados através da minimização da soma dos quadrados das diferenças entre o valor estimado e os dados observados.

A equação (2.5) pode ser representada na forma de erro de predição:

$$y(k) = \sum_{i=1}^{n_\theta} \varphi_i^T(k) \hat{\theta}_i + \xi(k) \quad (2.11)$$

onde  $n_\theta$  é o número de termos no modelo, o símbolo “^” sobre variáveis indica valores estimados,  $\varphi(k)$  é o vetor de regressores,  $\hat{\theta}$  indica os parâmetros a serem estimados e  $\xi(k)$  representa os resíduos., sendo definido como:

$$\xi(k) = y(k) - \hat{y}(k) \quad (2.12)$$

onde  $\hat{y}(k)$  é a predição de um passo à frente de  $y(k)$ , dada por:

$$\hat{y}(k) = \sum_{i=1}^{n_\theta} \varphi_i^T(k) \hat{\theta}_i \quad (2.13)$$

Admitindo que sejam realizadas  $N$  medições, suficientes para determinar os parâmetros, tem-se:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_{n_\theta}(1) \\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_{n_\theta}(2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(N) & \varphi_2(N) & \cdots & \varphi_{n_\theta}(N) \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_{n_\theta} \end{bmatrix} + \begin{bmatrix} \xi(1) \\ \xi(2) \\ \vdots \\ \xi(N) \end{bmatrix} \quad (2.14)$$

A equação (2.14) pode ser escrita na forma matricial, tornando-se:

$$Y = \Psi \hat{\Theta} + \Xi \quad (2.15)$$

onde  $\Psi$  é a matriz de regressores,  $\hat{\Theta}$  é o vetor de parâmetros a serem estimados e  $\Xi$  é o vetor de resíduos. Da equação (2.15), verifica-se que  $\Xi = Y - \Psi\hat{\Theta}$ , ou seja, o vetor de resíduos é composto pela diferença entre o vetor de saídas observadas do sistema  $Y$  e a estimativa desse valor a partir dos dados de estimação  $\Psi$  usando o vetor de parâmetros estimado  $\hat{\Theta}$ .

O objetivo, então, é encontrar  $\hat{\Theta}$  que minimize a função de custo do algoritmo MQ, dada por:

$$J_{MQ} = \sum_{i=1}^N \xi(k)^2 = \Xi^T \Xi \quad (2.16)$$

A solução que minimiza a equação (2.16) é o estimador MQ, definido por:

$$\hat{\Theta} = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (2.17)$$

A matriz  $\Psi^T \Psi$  é denominada matriz de informação ou matriz de covariância dos regressores. Esta matriz é simétrica e definida positiva quando  $\Psi$  tem posto pleno de colunas e semidefinida positiva em caso contrário. Entretanto, ela pode perder essa propriedade se estiver excessivamente mal-condicionada. A solução  $\hat{\Theta}$  da equação (2.17) existe e é única desde que  $\Psi^T \Psi$  seja não singular (AGUIRRE; RODRIGUES; JÁCOME, 1998).

## 2.6 VALIDAÇÃO DE MODELOS

A validação de modelos é uma parte fundamental no estudo de identificação de sistemas e consiste em verificar se os modelos obtidos são representações adequadas para o sistema considerado, ou seja, deve-se verificar se o modelo é válido ou não. É importante ressaltar que nenhum modelo irá representar o sistema físico em todos os aspectos. Desta forma, um modelo pode ser considerado válido se incorporar algumas características do sistema consideradas fundamentais para a aplicação desejada. Vale frisar, ainda, que procedimento de validação deve ser independente do tipo de modelo e dos métodos utilizados para obtê-los (AGUIRRE,



2015; BILLINGS, 2013). A seguir, serão apresentados os métodos de validação empregados neste trabalho.

### 2.6.1 Simulação

Simular o modelo obtido e comparar o resultado com os dados medidos é a maneira mais usual de validação. Neste procedimento, o objetivo é verificar se o modelo reproduz ao longo do tempo os dados observados. Trata-se de uma maneira simples e bastante comum de efetuar a validação. Contudo, deve-se atentar para não utilizar na validação os mesmos dados aplicados na obtenção do modelo, com o propósito de verificar a capacidade de generalização do mesmo. Para isto, testes independentes devem ser efetuados para gerar um conjunto de dados de identificação e outro conjunto de validação. Caso não seja viável realizar testes distintos, deve-se dividir o conjunto de dados disponível em duas partes (não necessariamente iguais), sendo a primeira usada para identificação e a segunda para validação (AGUIRRE, 2015).

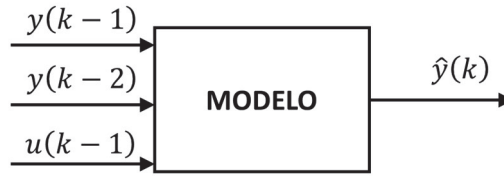
#### 2.6.1.1 Predição de um passo à frente

É possível simular um modelo de maneiras diferentes, sendo uma delas a predição de um passo à frente. Neste método, para se obter o valor predito no instante  $k$ , deve-se aplicar observações até o instante  $(k - 1)$  no modelo e calcular a sua saída. Seja o seguinte modelo, considerado a título de exemplo:

$$y(k) = \theta_1 y(k - 1) + \theta_2 y(k - 2) + \theta_3 u(k - 1) + e(k) \quad (2.18)$$

Para este exemplo, a predição de um passo à frente pode ser representada pelo diagrama da FIGURA 2, onde é possível observar que a saída predita,  $\hat{y}(k)$ , é calculada com base em valores passados observados tanto da entrada,  $u(k)$ , quanto da saída  $y(k)$ .

FIGURA 2 – Diagrama do processo de predição de um passo à frente



FONTE: O autor (2020).

A predição de um passo à frente, a partir do instante  $k = 3$  é obtida da seguinte forma:

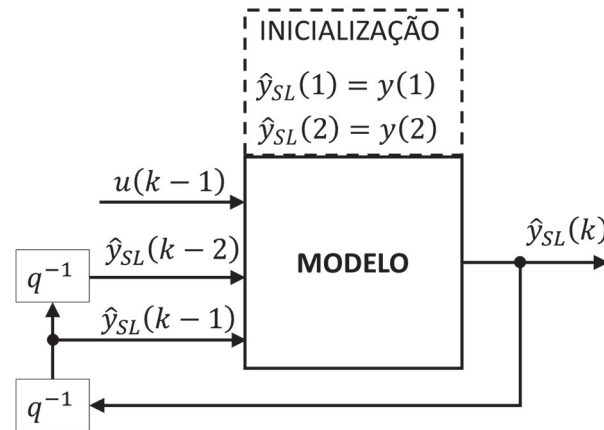
$$\begin{cases} \hat{y}(3) = \hat{\theta}_1 y(2) + \hat{\theta}_2 y(1) + \hat{\theta}_3 u(2) \\ \hat{y}(4) = \hat{\theta}_1 y(3) + \hat{\theta}_2 y(2) + \hat{\theta}_3 u(3) \\ \hat{y}(5) = \hat{\theta}_1 y(4) + \hat{\theta}_2 y(3) + \hat{\theta}_3 u(4) \\ \vdots \\ \hat{y}(k) = \hat{\theta}_1 y(k-1) + \hat{\theta}_2 y(k-2) + \hat{\theta}_3 u(k-1) \\ \vdots \end{cases} \quad (2.19)$$

A partir da equação (2.19), observa-se que a predição de um passo à frente em um instante posterior,  $\hat{y}(k+1)$ , não utilizará a predição do instante atual,  $\hat{y}(k)$ . Utilizar a predição de um passo à frente é bastante conveniente do ponto de vista numérico, porém deve-se atentar para o fato de que uma boa predição de um passo à frente não garante que o modelo seja válido do ponto de vista dinâmico e alternativas adicionais de validação devem ser exploradas (AGUIRRE, 2015).

#### 2.6.1.2 Simulação livre

Outra maneira de simular um modelo é através da simulação livre, também chamada de predição de infinitos passos à frente ou predição livre. Neste tipo de predição, a saída do sistema é inicializada com valores de medidas conhecidos e então a simulação livre é calculada com base em valores observados da entrada e valores preditos da saída (BILLINGS, 2013). Para o sistema utilizado como exemplo na equação (2.18), a simulação livre pode ser representada pelo diagrama da FIGURA 3, onde  $\hat{y}_{SL}(k)$  é a saída predita através de simulação livre e  $q^{-1}$  representa o operador de atraso, de forma que  $\hat{y}_{SL}(k)q^{-1} = \hat{y}_{SL}(k-1)$ .

FIGURA 3 – Diagrama do processo de simulação livre



FONTE: O autor (2020).

Neste caso, a simulação livre é calculada da seguinte maneira:

$$\begin{cases} \hat{y}_{SL}(1) = y(1) \\ \hat{y}_{SL}(2) = y(2) \\ \hat{y}_{SL}(3) = \hat{\theta}_1 \hat{y}_{SL}(2) + \hat{\theta}_2 \hat{y}_{SL}(1) + \hat{\theta}_3 u(2) \\ \hat{y}_{SL}(4) = \hat{\theta}_1 \hat{y}_{SL}(3) + \hat{\theta}_2 \hat{y}_{SL}(2) + \hat{\theta}_3 u(3) \\ \hat{y}_{SL}(5) = \hat{\theta}_1 \hat{y}_{SL}(4) + \hat{\theta}_2 \hat{y}_{SL}(3) + \hat{\theta}_3 u(4) \\ \vdots \\ \hat{y}_{SL}(k) = \hat{\theta}_1 \hat{y}_{SL}(k-1) + \hat{\theta}_2 \hat{y}_{SL}(k-2) + \hat{\theta}_3 u(k-1) \\ \vdots \end{cases} \quad (2.20)$$

Como pode ser observado na equação (2.20), naturalmente a simulação livre utilizará nas primeiras iterações uma mistura de valores medidos e preditos da variável de saída  $y$ . Diferentemente da predição de um passo à frente, a simulação livre é mais adequada para verificar se o modelo se ajusta dinamicamente aos dados do sistema. Vale ressaltar que, ao simular modelos para validação, o sinal de entrada usado sempre será o sinal de entrada medido (AGUIRRE, 2015).

### 2.6.1.3 Medidas de desempenho

Uma maneira de analisar os dados gerados pelas predições é através de gráficos, onde será possível fazer uma verificação visual do ajuste entre a saída do

modelo e a saída coletada do sistema. Contudo, é útil também fazer uso de medidas quantitativas dessa aproximação através de determinados índices. Uma medida de desempenho frequentemente utilizada em validação de modelos é a raiz quadrada do erro médio quadrático (RMSE, do inglês *Root Mean Square Error*) calculado por meio da seguinte equação (PEÑARANDA; SAAVEDRA-MONTES, 2012):

$$RMSE = \sqrt{\frac{\sum_{k=1}^N (y(k) - \hat{y}(k))^2}{N}} \quad (2.21)$$

onde  $N$  é o total de amostras considerado. Neste caso o  $RMSE$  é calculado com base na diferença  $y(k) - \hat{y}(k)$ , chamada de erro de predição. Quanto menor o  $RMSE$ , maior é a aproximação da predição aos dados medidos. Também é possível utilizar a diferença  $y(k) - \hat{y}_{SL}(k)$ , chamada de erro de simulação, para calcular o  $RMSE$ , que neste caso será denotado por  $RMSE_{SL}$ .

Outro índice bastante utilizado para validar dados de predição é o coeficiente de determinação ( $R^2$ ), dado por (CAMERON; WINDMEIJER, 1996):

$$R^2 = 1 - \frac{\sum_{k=1}^N (y(k) - \hat{y}(k))^2}{\sum_{k=1}^N (y(k) - \bar{y}(k))^2} \quad (2.22)$$

onde  $\bar{y}(k)$  é a média do sinal de saída observado. O coeficiente de determinação apresenta valores no intervalo  $[0,1]$ , podendo também ser expresso em porcentagem. Quanto mais próximo de 1 (ou 100%), melhor é a aderência da saída gerada pelo modelo à saída real. Para os casos em que seja utilizado  $\hat{y}_{SL}(k)$  no lugar de  $\hat{y}(k)$ , será adotada a notação  $R_{SL}^2$ .

### 2.6.2 Validação estatística

Realizar a validação por simulação é um procedimento simples e bastante útil, porém apresenta algumas limitações. Ao analisar uma simulação, caso ocorram diferença significativas entre as predições e os dados observados, não é possível

afirmar se o problema está na estimação de parâmetros ou se a dinâmica que produziu os dados não pode ser representada pelo modelo de maneira satisfatória (AGUIRRE, 2015). Uma alternativa que ajuda a contornar esta problemática é a validação estatística ou análise de resíduos. Este tipo de análise é baseado no fato de que os resíduos não devem ser autocorrelacionados ou correlacionados com qualquer um dos regressores (BILLINGS; VOON, 1983). Considerando-se que os resíduos compõem a parte dos dados que o modelo não conseguiu explicar, espera-se que estes tenham o comportamento de um ruído branco. Em outras palavras, do ponto de vista da validação de modelos, é desejável que os resíduos sejam aleatórios e não contenham informação útil acerca dos dados (AGUIRRE, 2015).

Para realizar a análise de resíduos, inicialmente deve-se definir a função de correlação cruzada  $\phi_{wz}(\tau)$  entre dois sinais  $w$  e  $z$ , como sendo:

$$\phi_{wz}(\tau) = \frac{\sum_{k=1}^{N-\tau} ([w(k) - \bar{w}][z(k + \tau) - \bar{z}])}{\sqrt{\sum_{k=1}^N [w(k) - \bar{w}]^2} \sqrt{\sum_{k=1}^N [z(k) - \bar{z}]^2}} \quad (2.23)$$

onde a barra superior indica o valor médio da variável em que esta se encontra (BILLINGS, 2013).

Com o vetor de resíduos definido na equação (2.12), calculado sobre os dados de estimação, utiliza-se o seguinte conjunto de funções (BILLINGS; VOON, 1986):

$$\begin{cases} \phi_{\xi\xi}(\tau) = \delta(\tau), & \forall \tau \\ \phi_{u\xi}(\tau) = 0, & \forall \tau \\ \phi_{\xi(\xi u)}(\tau) = 0, & \tau \geq 0 \\ \phi_{(u^2)'\xi}(\tau) = 0, & \forall \tau \\ \phi_{(u^2)'\xi^2}(\tau) = 0, & \forall \tau \end{cases} \quad (2.24)$$

onde  $\delta(\tau)$  é o impulso unitário,  $(u^2)'(k) = u^2(k) - E[u^2(k)] = u^2(k) - \bar{u}^2$ , e  $(\xi u)(k) = \xi(k + 1)u(k + 1)$ . No conjunto de equações (2.24), as duas primeiras funções são as funções de autocorrelação e correlação cruzada (lineares) e as demais relações são não lineares. Na prática, estas funções são aplicadas de maneira normalizada e analisadas em conjunto com o intervalo de confiança de 95%, o qual é

definido como o valor aproximado de  $\pm 1,96/\sqrt{N}$ , onde  $N$  é o número de amostras. Para esta análise deve-se observar se algum valor das funções, com exceção da primeira, encontra-se fora da faixa de confiança para os primeiros valores de  $\tau$  (tipicamente até  $\tau = 5$ ) (BILLINGS, 2013).

Vale ressaltar que o conjunto de equações (2.24) não esgota todas as possibilidades de análise de resíduos. Existem outros testes como, por exemplo, os testes de correlação envolvendo os resíduos,  $\xi(k)$ , a entrada,  $u(k)$ , e a saída,  $y(k)$ , apresentados por Billings e Zhu (1994).

## 2.7 RESUMO DO CAPÍTULO

Neste capítulo foram apresentados os principais conceitos relacionados à identificação de sistemas, bem como o detalhamento de cada uma das etapas que a compõem. Foi abordada a importância da etapa de coleta de dados, frisando alguns aspectos principais, como os tipos de sinais de excitação, o período de amostragem e o cuidado que se deve ter em utilizar, na obtenção do modelo, dados distintos daqueles empregados na sua validação. A definição e embasamento matemático do modelo NARMAX foram explorados, além de evidenciar o seu subtipo que será utilizado neste trabalho, o modelo NARX. Algumas técnicas de detecção de estrutura foram abordadas, com destaque para a seleção por meio de metaheurística, a qual configura o ponto central deste trabalho. Foi apresentado, também, o método dos mínimos quadrados, que será aplicado na estimação de parâmetros dos modelos propostos para os estudos de caso desta pesquisa. Por fim, foram abordadas as validações por meio de simulações e análise de resíduos, que servirão como forma de avaliação dos modelos encontrados.

### 3 METAHEURÍSTICAS

#### 3.1 INTRODUÇÃO

A otimização é um importante campo de estudo, cujos conceitos podem ser aplicados na análise e projeto de sistemas nas mais diversas áreas, como engenharia, economia, medicina, gestão, dentre outras. No contexto de identificação de sistemas, conforme apresentado na seção 2.4, a etapa de seleção de estrutura pode ser definida como um problema de otimização. Desta forma, faz-se necessário a aplicação de técnicas adequadas para a solução deste tipo de problema.

Dentre as diversas técnicas de otimização, destacam-se as metaheurísticas, que formam uma classe de métodos aproximados, projetados para atacar problemas complexos de otimização. Metaheurísticas são definidas como um processo de geração iterativo que guia uma heurística subordinada, combinando diferentes conceitos e criando um processo capaz de escapar de ótimos locais e realizar uma busca robusta de um espaço de solução (BOUSSAÏD; LEPAGNOT; SIARRY, 2013; DU; SWAMY, 2016). Para resolver problemas complexos de otimização, a maioria das metaheurísticas imita metáforas naturais, como por exemplo, evolução de espécies, processo de recozimento, colônia de formigas, enxame de partículas, sistema imune, colônia de abelhas e enxame de vespas (SÖRENSEN; GLOVER, 2013).

Nas seções seguintes, serão dados detalhes a respeito das metaheurísticas aplicadas neste trabalho para solucionar o problema de seleção de estrutura de modelos polinomiais.

#### 3.2 ALGORITMO GENÉTICO

Algoritmo Genético (GA, do inglês *Genetic Algorithms*) é um método de busca e otimização baseado nos princípios da genética e da seleção natural, capaz de avaliar simultaneamente vários pontos no espaço de pesquisa e retornar a melhor solução ou um grupo das melhores soluções para o problema em questão (HAUPT; HAUPT, 2004).

Os principais trabalhos que serviram de inspiração para o desenvolvimento do GA foram: a publicação de Darwin de 1859 intitulada “A origem das espécies por meio da seleção natural”; e o trabalho de Mendel sobre o princípio básico de herança

genética, que foi redescoberto por volta de 1900 e teve grande influência sobre trabalhos relacionados à evolução. A partir desta combinação, muitos biólogos começaram a desenvolver simulações computacionais de Sistemas Genéticos nos anos de 1950 a 1960. Finalmente, em 1975 John Holland refinou suas ideias e publicou o livro hoje considerado referência básica de GA, intitulado “*Adaptation in Natural and Artificial Systems*” (HOLLAND, 1975). Contudo, o método somente foi popularizado por meio de um dos seus estudantes, David Goldberg (GOLDBERG, 1989).

O GA difere dos métodos tradicionais de busca e otimização, principalmente por possuir as seguintes características (GOLDBERG, 1989):

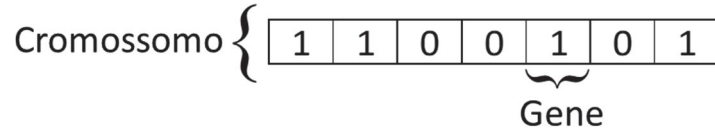
- trabalha com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- opera com uma população e não com um único ponto;
- utiliza informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;
- aplica regras de transição probabilísticas e não determinísticas.

A busca empregada pelo GA ocorre de modo paralelo e estruturado, porém é feita de maneira aleatória, voltada em direção ao reforço da busca de pontos de “alta aptidão”, ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos). Apesar de aleatória, essa busca é feita de forma direcionada permitindo explorar informações históricas para encontrar novos ponto de busca onde são esperados melhores desempenhos.

O processo de busca é feito através de passos iterativos, sendo cada iteração chamada de geração. Em cada iteração são aplicados princípios de seleção e reprodução a uma população de candidatos, chamados de cromossomos e que representam as possíveis soluções do problema, geralmente codificadas como uma cadeia de bits. Cada cromossomo é formado por genes, os quais podem ser cada bit individualmente ou blocos de bits adjacentes que codificam elementos particulares da solução candidata. Por meio da seleção serão determinados quais indivíduos estarão aptos a se reproduzir, através do cruzamento, gerando uma certa quantidade de descendentes para a próxima geração. Além disso, também é possível ocorrer alterações nos cromossomos através da mutação (MITCHELL, 1998). Um exemplo de cromossomo é apresentado na FIGURA 4.



FIGURA 4 – Cromossomo em que cada bit representa um gene



FONTE: O autor (2020).

O GA é inicializado com a criação de uma população de  $N_{P_G}$  cromossomos, ou seja, vetores com variáveis que formam possíveis soluções do problema. Se o problema possui  $N_{V_G}$  variáveis, então um cromossomo pode ser escrito como:

$$cromossomo = \{p_1, p_2, p_3, \dots, p_{N_{V_G}}\} \quad (3.1)$$

A cada cromossomo é associado um valor de custo, também chamado de aptidão, o qual é determinado pela aplicação de uma função objetivo,  $J_G$ , nas variáveis  $p_1, p_2, p_3, \dots, p_{N_{V_G}}$ :

$$aptidão = J_G(cromossomo) = J_G(p_1, p_2, p_3, \dots, p_{N_{var}}) \quad (3.2)$$

### 3.2.1 Seleção natural

Quando se aplica o GA em um problema de minimização, a sobrevivência do mais apto traduz-se no descarte de cromossomos com elevados valores de custo. Primeiro, os  $N_{P_G}$  valores de custo e seus respectivos cromossomos são classificados do menor para o maior. Em seguida, somente os melhores são selecionados para permanecer na população, enquanto os demais são excluídos. A taxa de seleção natural,  $t_{sn}$ , é a fração de  $N_{P_G}$  que sobrevive para as próximas etapas do processo. Normalmente adota-se uma taxa de seleção de 50% (HAUPT; HAUPT, 2004). O número de cromossomos sobreviventes, que são mantidos a cada geração, é dado por:

$$N_{sob} = t_{sn} \cdot N_{P_G} \quad (3.3)$$

A seleção natural ocorre a cada geração ou iteração do GA. Dos  $N_{P_G}$  cromossomos iniciais em uma geração, somente  $N_{sob}$  sobrevivem para as etapas seguintes e  $N_{P_G} - N_{sob}$  cromossomos são excluídos para dar espaço à criação de novos descendentes, através de seleção para reprodução, cruzamento e mutação.

### 3.2.2 Seleção para reprodução

Na seleção para reprodução, ocorre a formação de pares que irão dar origem a novos descendentes. Dois cromossomos pais, selecionados dentro do conjunto  $N_{sob}$ , irão gerar dois cromossomos filhos. Novos pares de cromossomos pais vão sendo formados até que a quantidade de cromossomos filhos seja suficiente para repor os  $N_{P_G} - N_{sob}$  cromossomos descartados na seleção natural (HAUPT; HAUPT, 2004). Alguns dos principais métodos para selecionar pares para reprodução são descritos a seguir.

#### 3.2.2.1 Seleção por emparelhamento do melhor ao pior

Partindo do início da lista de cromossomos em  $N_{sob}$ , os elementos são agrupados dois a dois, até que a quantidade necessária seja atingida. Nos pares formados, cada cromossomo numa posição ímpar representa uma mãe e cada cromossomo numa posição par representa um pai. Esta abordagem não imita bem o comportamento da natureza, mas apresenta a vantagem de ser bem simples de implementar.

#### 3.2.2.2 Seleção por roleta

A probabilidade de um cromossomo participar da reprodução é inversamente proporcional ao seu custo. Cromossomos com menores custos apresentam maiores probabilidades de se reproduzirem e repassarem seus genes adiante. Para determinar a probabilidade de reprodução,  $P_j$ , calcula-se o custo normalizado de cada

cromossomo subtraindo o menor custo presente no grupo de cromossomos excluídos na seleção natural,  $c_{N_{sob}+1}$ , de cada custo presente no grupo  $N_{sob}$ :

$$C_j = c_j - c_{N_{sob}+1} \quad (3.4)$$

Na equação (3.4),  $c_j$  é o custo do cromossomo na posição  $j$  e  $C_j$  é seu respectivo custo normalizado. Desta forma,  $P_j$  é dada por:

$$P_j = \left| \frac{C_j}{\sum_{m=1}^{N_{sob}} C_m} \right| \quad (3.5)$$

Utiliza-se, então, a probabilidade acumulada de cada cromossomo,  $\sum_{i=1}^j P_i$  para realizar o processo de seleção. Primeiro, é gerado um número aleatório no intervalo  $[0,1]$ , a partir de uma distribuição uniforme. Em seguida, partindo-se do início da lista de cromossomos em  $N_{sob}$ , o primeiro cromossomo que possuir probabilidade acumulada maior que o número aleatório gerado é selecionado para participar da reprodução. O processo é repetido até que sejam selecionados cromossomos suficientes para formar a quantidade de pares necessária.

#### 3.2.2.3 Seleção por torneio

Neste método, pequenos subconjuntos de cromossomos (dois ou três) são formados a partir de indivíduos selecionados aleatoriamente do conjunto  $N_{sob}$ , a partir de uma distribuição uniforme. Os cromossomos de um subconjunto têm seus custos comparados e aquele que possuir o menor custo é selecionado para reprodução. O processo se repete até que a quantidade necessária de pares seja alcançada.

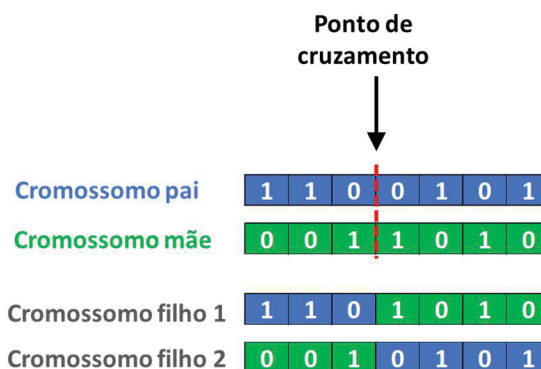
#### 3.2.3 Cruzamento

O cruzamento consiste em criar um ou mais filhos a partir dos cromossomos pais escolhidos no processo de seleção para reprodução. Desta forma, o material genético dos indivíduos mais aptos da população é propagado, imitando

aproximadamente o que ocorre na recombinação biológica (MITCHELL, 1998). Com o cruzamento também é possível explorar regiões ainda não conhecidas no espaço de busca.

A maneira mais comum de efetuar o cruzamento é quando um par de cromossomos pais gera um par de cromossomos filhos (HAUPT; HAUPT, 2004). Neste processo, inicialmente é selecionado um ponto de cruzamento aleatoriamente, com base em uma distribuição uniforme, entre o primeiro e o último bit dos cromossomos pais. Em seguida, faz-se a recombinação de genes da seguinte forma: os bits do cromossomo pai que estão à esquerda do ponto de cruzamento combinam-se com os bits do cromossomo mãe que estão à direita do ponto de cruzamento, formando o primeiro cromossomo filho. Um segundo cromossomo filho também é formado, combinando-se os bits do cromossomo pai que estão à direita do ponto de cruzamento com os bits do cromossomo mãe que estão à esquerda do ponto de cruzamento (HAUPT; HAUPT, 2004). Um exemplo de cruzamento com um ponto está ilustrado na FIGURA 5.

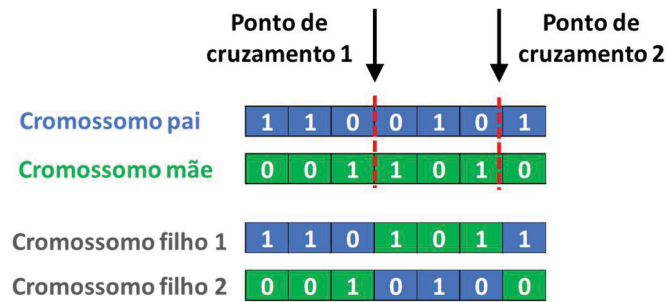
FIGURA 5 – Cruzamento de um único ponto



FONTE: O autor (2020).

Também é possível aplicar dois ou mais pontos de cruzamento para gerar cromossomos filhos, fazendo a combinação de material genético de maneira semelhante ao caso de um único ponto. Na FIGURA 6 é apresentado um exemplo de cruzamento com dois pontos, gerando dois cromossomos filhos.

FIGURA 6 – Cruzamento de dois pontos



FONTE: O autor (2020).

Normalmente, o cruzamento não é aplicado a todos os pares de indivíduos selecionados para reprodução. Aplica-se uma probabilidade de se utilizar o cruzamento, denominada taxa de cruzamento,  $t_c$ , tipicamente na faixa entre 0,6 e 0,9. Para determinar se o cruzamento irá ocorrer, é gerado um número aleatório por uma distribuição uniforme e, se este for menor que  $t_c$ , o cruzamento ocorre normalmente. Caso contrário, os cromossomos filhos simplesmente permanecem iguais aos cromossomos pais (BEASLEY; BULL; MARTIN, 1993).

### 3.2.4 Mutação

Após a geração dos cromossomos filhos, o GA executa a operação de mutação. Esta tem por finalidade introduzir sobre os cromossomos recém gerados material genético diferente daqueles provenientes dos cromossomos pais, possibilitando a busca em regiões ainda não exploradas e evitando a convergência prematura para mínimos locais. Quando se trabalha com codificação binária, a mutação é feita alterando-se um bit 0 para 1 ou vice-versa. A decisão se determinado bit será alterado é feita com base na probabilidade de mutação,  $p_m$ , cujos valores típicos encontram-se entre 0,001 e 0,05. Para isto, gera-se um número aleatório a partir de uma distribuição uniforme e, se este for menor que  $p_m$ , o bit sofre mutação; caso contrário, permanece com seu valor original. Os bits de cada cromossomo são verificados de maneira independente, ou seja, a mudança em um bit não afeta a probabilidade de a mutação ocorrer em um bit distinto (SRINIVAS; PATNAIK, 1994).

Após realizar todos estes passos (seleção natural, seleção para reprodução, cruzamento e mutação), completa-se uma iteração, ou geração, do GA. Estas

operações são repetidas até que algum critério de parada seja atendido, como por exemplo o número  $g_G$  de gerações atingir um valor máximo  $g_{G_{max}}$ . Ao final, o indivíduo que possuir o melhor valor de função objetivo será selecionado como solução do problema.

É possível encontrar na literatura diversos estudos que aplicam versões modificadas do GA, projetadas para melhorar seu desempenho, principalmente através de alterações nos seus operadores (CHIANG, 2005; DANDY; SIMPSON; MURPHY, 1996; LEUNG et al., 2003; LIU; TENG, 1999; PARK et al., 2000; SILVA; GIL; AREIZA, 2000). Todavia, neste trabalho será aplicada a versão básica que foi detalhada nesta seção.

### 3.3 EVOLUÇÃO DIFERENCIAL

A Evolução Diferencial (DE, do inglês *Differential Evolution*) foi criada após Ken Price tentar usar técnicas evolutivas para resolver o problema do polinômio de Chebychev, o qual foi apresentado por Rainer Storn. Esta descoberta ocorreu quando Price tentou utilizar diferentes vetores para recriar o vetor população dando origem a uma abordagem diferente daquelas encontradas nas estratégias de evolução que tratam de problemas de otimização.

A inicialização do algoritmo é feita por meio da geração de uma população de  $N_{P_D}$  indivíduos, escolhida aleatoriamente a partir de uma distribuição uniforme e devendo cobrir todo o espaço de busca. Quando não há nenhum conhecimento sobre o problema, é comum criar a população inicial por meio de uma distribuição uniforme (PRICE, 1996).

Os indivíduos, chamados de vetores, possuem tamanho  $N_{V_D}$  e contém as variáveis que formam as possíveis soluções do problema tratado. A população evolui ao longo de um determinado número de gerações e, a cada geração, a quantidade de indivíduos na população permanece constante. Cada indivíduo é representado por  $\mathbf{x}_{j,g_D} = (x_{i,j,g_D})$ , onde  $i = 1, \dots, N_{V_D}$  são os índices dos parâmetros,  $j = 1, \dots, N_{P_D}$  representa o índice do indivíduo dentro da população e  $g_D = 0, 1, \dots, g_{D_{max}}$  indica a geração à qual o indivíduo pertence.

Para gerar novos indivíduos, são aplicados na população os operadores de mutação, cruzamento e seleção. Na mutação é feita a adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo, gerando o chamado vetor mutante. Na etapa seguinte, chamada de cruzamento, o vetor mutante tem suas componentes misturadas com as componentes de um vetor escolhido aleatoriamente na população (vetor alvo), gerando, desta forma, o vetor tentativa. Por fim, na etapa de seleção, é verificado se o vetor tentativa possui um valor de função objetivo menor que o vetor alvo e, em caso afirmativo, o vetor tentativa ocupa o lugar do vetor alvo na geração seguinte. Este ciclo se repete até que algum critério de parada seja atingido, como por exemplo, o número de gerações chegar a um valor máximo  $g_{D_{max}}$  (STORN; PRICE, 1997).

### 3.3.1 Mutação

Após a inicialização, a DE cria um vetor mutante  $\mathbf{v}_{j,g_D}$ , para cada vetor alvo  $\mathbf{x}_{j,g_D}$ ,  $j = 1, \dots, N_{P_D}$ , na geração corrente, através da mutação. A formulação básica para a mutação é dada por:

$$\mathbf{v}_{j,g_D} = \mathbf{x}_{d_0,g_D} + F_m \cdot (\mathbf{x}_{d_1,g_D} - \mathbf{x}_{d_2,g_D}) \quad (3.6)$$

Na equação (3.6),  $\mathbf{x}_{d_0,g_D}$  é o vetor base e  $(\mathbf{x}_{d_1,g_D} - \mathbf{x}_{d_2,g_D})$  é chamado de vetor diferença. O fator de escala,  $F_m$ , é um número real positivo que controla a taxa com que a população evolui. Não existe um limite exato para  $F_m$ , porém, valores efetivos raramente serão maiores do que 1 (PRICE; STORN; LAMPINEN, 2005).

O índice do vetor base,  $d_0$ , deve ser diferente do índice do vetor alvo,  $j$ , e pode ser selecionado de várias maneiras, sendo mais comum efetuar uma escolha aleatória (*rand*) a partir de uma distribuição uniforme, como apresentado na equação (3.6), ou utilizar o melhor indivíduo encontrado até a geração corrente (*best*), conforme a equação (3.7).

$$\mathbf{v}_{j,g_D} = \mathbf{x}_{best} + F_m \cdot (\mathbf{x}_{d_1,g_D} - \mathbf{x}_{d_2,g_D}) \quad (3.7)$$

Os índices do vetor diferença,  $d_1$  e  $d_2$ , são escolhidos aleatoriamente com base em uma distribuição uniforme e devem diferir um do outro e do índice do vetor alvo,  $j$ .

A mutação também pode ser realizada utilizando dois vetores diferença, como pode ser observado nas equações (3.8) e (3.9).

$$\mathbf{v}_{j,g_D} = \mathbf{x}_{d_0,g_D} + F_m \cdot (\mathbf{x}_{d_1,g_D} - \mathbf{x}_{d_2,g_D}) + F_m \cdot (\mathbf{x}_{d_3,g_D} - \mathbf{x}_{d_4,g_D}) \quad (3.8)$$

$$\mathbf{v}_{j,g_D} = \mathbf{x}_{best} + F_m \cdot (\mathbf{x}_{d_1,g_D} - \mathbf{x}_{d_2,g_D}) + F_m \cdot (\mathbf{x}_{d_3,g_D} - \mathbf{x}_{d_4,g_D}) \quad (3.9)$$

### 3.3.2 Cruzamento

Para aumentar a diversidade potencial da população, após a mutação é realizado o cruzamento. Nesta etapa, é criado um vetor tentativa,  $\mathbf{u}_{j,g_D}$ , formado pela permuta entre as componentes do vetor mutante e do vetor alvo, conforme visto na equação (3.10).

$$\mathbf{u}_{j,g_D} = u_{i,j,g_D} = \begin{cases} v_{i,j,g_D} & \text{se } rand_i(0,1) \leq Cr \\ x_{i,j,g_D} & \text{se } rand_i(0,1) > Cr \end{cases} \quad (3.10)$$

A probabilidade de cruzamento,  $Cr \in [0,1]$ , é um parâmetro que controla a proporção de componentes que passarão do vetor mutante para o vetor tentativa.

No cruzamento apresentado na equação (3.10), chamado de binomial, para cada componente  $i$  do vetor tentativa  $\mathbf{u}_{j,g_D} = u_{i,j,g_D}$ , é gerado um número aleatório  $rand_i$ , a partir de uma distribuição uniforme, que irá determinar se  $\mathbf{u}_{j,g_D}$  receberá a  $i$ -ésima componente do vetor mutante,  $\mathbf{v}_{j,g_D}$ , ou do vetor alvo,  $\mathbf{x}_{j,g_D}$ . Outro tipo de cruzamento utilizado é o exponencial, onde o vetor tentativa inicia recebendo as componentes do vetor mutante e este processo se repete enquanto  $rand_i \leq Cr$ . Na primeira ocorrência de  $rand_i > Cr$  o vetor tentativa passará a receber as componentes do vetor alvo (PRICE; STORN; LAMPINEN, 2005).



### 3.3.3 Seleção

Para manter o tamanho da população constante a cada geração subsequente, é realizada a etapa de seleção, a qual determina se o vetor alvo ou o vetor tentativa permanecerá na população na geração seguinte. A operação de seleção é descrita como:

$$\mathbf{x}_{j,g_D+1} = \begin{cases} \mathbf{u}_{j,g_D} & \text{se } J_D(\mathbf{u}_{j,g_D}) \leq J_D(\mathbf{x}_{j,g_D}) \\ \mathbf{x}_{j,g_D} & \text{se } J_D(\mathbf{u}_{j,g_D}) > J_D(\mathbf{x}_{j,g_D}) \end{cases} \quad (3.11)$$

onde  $J_D(\cdot)$  é a função objetivo a ser minimizada pela DE. Desta forma, se vetor tentativa,  $\mathbf{u}_{j,g_D}$ , tiver um valor de função objetivo menor ou igual que o de seu vetor alvo,  $\mathbf{x}_{j,g_D}$ , ele ocupa seu lugar na próxima geração. Caso contrário, o vetor alvo mantém sua posição na população por pelo menos mais uma geração.

Quando este ciclo é concluído para uma geração, as etapas de mutação, cruzamento e seleção são repetidas até que o valor ótimo de  $J_D(\cdot)$  seja localizado ou algum outro critério de parada seja atingido. Um critério comumente adotado é a quantidade máxima de gerações,  $g_{D_{max}}$ , mas também podem ser considerados como critérios de parada, por exemplo, a não violação de alguma restrição do problema, a taxa de decaimento do valor da função objetivo, dentre outros.

### 3.3.4 Estratégias

Cada maneira de realizar a mutação e o cruzamento configura uma estratégia da DE. As diversas estratégias são representadas pela notação DE/ $s_1$ / $s_2$ / $s_3$ , onde:  $s_1$  representa o tipo de vetor base;  $s_2$  é a quantidade de vetores diferença; e  $s_3$  indica o tipo de cruzamento. Por exemplo, a mutação apresentada na equação (3.6) combinada com o cruzamento binomial formam a estratégia DE/rand/1/bin, também conhecida como estratégia clássica. A TABELA 1 reúne as estratégias apresentadas até aqui.

TABELA 1 – Estratégias da Evolução Diferencial

Vetor Base	Vetores Diferença	Cruzamento	Notação
Aleatório	1	Binomial	DE/rand/1/bin
<i>Best</i>	1	Binomial	DE/best/1/bin
Aleatório	2	Binomial	DE/rand/2/bin
<i>Best</i>	2	Binomial	DE/best/2/bin
Aleatório	1	Exponencial	DE/rand/1/exp
<i>Best</i>	1	Exponencial	DE/best/1/exp
Aleatório	2	Exponencial	DE/rand/2/exp
<i>Best</i>	2	Exponencial	DE/best/2/exp

FONTE: O autor (2020).

Diversos estudos apresentam versões melhoradas da DE, seja através de modificações nos operadores de mutação, cruzamento e seleção (CHIANG; LEE; HEH, 2010; GUAN, 2016; WANG; CAI; ZHANG, 2012; ZHOU et al., 2016), seja por meio de métodos autoadaptativos para o fator de escala e a probabilidade de cruzamento (AL-DABBAGH, 2015; BREST et al., 2006; ZHANG; SANDERSON, 2007; LIU; LAMPINEN, 2005; QIN; SUGANTHAN, 2005). Entretanto, nesta pesquisa serão utilizadas versões básicas da DE, com parâmetros fixos e algumas das estratégias presentes na TABELA 1.

### 3.4 ALGORITMO DO MORCEGO

O Algoritmo do Morcego (BA, do inglês *Bat Algorithm*) é inspirado na capacidade de ecolocalização usada por morcegos durante seus movimentos de voo. Esta característica permite que os morcegos detectem suas presas, evitem obstáculos e localizem entradas de cavernas no escuro. Os morcegos emitem um pulso sonoro muito alto e ouvem o eco que retorna dos objetos circundantes. Seus pulsos possuem característica que podem variar e podem ser correlacionadas com suas estratégias de caça, dependendo da espécie. A maioria dos morcegos usa sinais curtos, modulados em frequência, para percorrer uma oitava, enquanto outros costumam usar sinais de frequência constante para ecolocalização. A largura de banda do sinal varia de acordo com a espécie e, geralmente, por causa dela os morcegos possuem a capacidade de detectar presas e evitar obstáculos na total escuridão (YANG, 2010).

No BA a inicialização é feita através da criação de um grupo de  $N_{PB}$  morcegos, que serão escolhidos aleatoriamente a partir de uma distribuição uniforme, e deverão conter, cada um, as  $N_{VB}$  variáveis que formam as possíveis soluções do

problema. Cada morcego voa aleatoriamente com velocidade  $\mathbf{a}_{j,g_B} = (a_{i,j,g_B})$  na posição  $\mathbf{b}_{j,g_B} = (b_{i,j,g_B})$ , onde  $i = 1, \dots, N_{V_B}$  são os índices das variáveis do problema,  $j = 1, \dots, N_{P_B}$  representa o índice do morcego dentro do grupo e  $g_B = 0, 1, \dots, g_{B_{max}}$  indica a iteração em que o algoritmo se encontra. As posições dos morcegos,  $\mathbf{b}_{j,g_B}$ , representam as possíveis soluções do problema e o propósito do BA é localizar aquela solução que minimiza a função objetivo  $J_B(\cdot)$ .

Para procurar suas presas, os morcegos emitem pulsos de frequência  $f_{j,g_B}$ , que pode variar entre um valor mínimo,  $f_{min}$ , e um máximo,  $f_{max}$ , previamente determinados. Os pulsos emitidos possuem amplitude  $H_{j,g_B}$ , que tende a reduzir a cada iteração do BA até atingir um valor mínimo, por exemplo  $H_{j,g_B} = 0$ , indicando que um determinado morcego localizou a presa e temporariamente parou a emissão de pulsos sonoros. A quantidade de pulsos emitidos a cada segundo é determinada pela taxa de emissão de pulsos,  $\omega_{j,g_B} \in [0,1]$ , a qual pode ser ajustada automaticamente dependendo da proximidade do alvo (YANG; GANDOMI, 2012).

A cada iteração, são realizadas atualizações nas posições e velocidades dos morcegos, bem como nas frequências, amplitudes e taxas de emissão dos pulsos sonoros até que algum critério de parada seja atingido, como por exemplo, o número de iterações chegar a um valor máximo  $g_{B_{max}}$  (YANG, 2010).

### 3.4.1 Movimentação dos morcegos

O movimento dos morcegos no BA é realizado através da atualização das suas posições e velocidades, a cada iteração, conforme as equações a seguir :

$$f_{B_j} = f_{B_{min}} + (f_{B_{max}} - f_{B_{min}}) \cdot h_B \quad (3.12)$$

$$\mathbf{a}_{j,g_B+1} = \mathbf{a}_{j,g_B} + (\mathbf{b}_{j,g_B} - \mathbf{b}_{g_B}^*) \cdot f_{B_j} \quad (3.13)$$

$$\mathbf{b}_{j,g_B+1} = \mathbf{b}_{j,g_B} + \mathbf{a}_{j,g_B+1} \quad (3.14)$$

onde  $h_B$  é um número aleatório escolhido com base em uma distribuição uniforme no intervalo  $[0,1]$  e  $\mathbf{b}_{g_B}^*$  é a melhor solução encontrada na iteração  $g_B$ , a qual é obtida após comparar as posições dos  $N_{P_B}$  morcegos (YANG; HE, 2013).

O BA pode realizar buscas locais em torno da melhor solução corrente,  $\mathbf{b}_{g_B}^*$ , permitindo que as demais soluções sejam melhoradas. Para isto, compara-se o valor de um número aleatório,  $rand_j \in [0,1]$ , gerado a partir de uma distribuição uniforme, com o valor da taxa de emissão de pulsos,  $\omega_{j,g_B}$ , de cada morcego. Se  $rand_j > \omega_{j,g_B}$ , realiza-se uma busca em torno da melhor solução atual, conforme a equação (3.15).

$$\mathbf{b}_{j,g_B}^* = \mathbf{b}_{g_B}^* + \varepsilon \bar{H}_{g_B} \quad (3.15)$$

onde  $\mathbf{b}_{j,g_B}^*$  é o valor da solução  $\mathbf{b}_{j,g_B}$  atualizado,  $\varepsilon \in [-1,1]$  é um número gerado aleatoriamente a partir de uma distribuição uniforme e  $\bar{H}_{g_B}$  é a média das amplitudes de emissão de todos os morcegos na iteração  $g_B$  (YANG; GANDOMI, 2012).

#### 3.4.2 Amplitude de pulsos e taxa de emissão dos pulsos

A amplitude de emissão dos pulsos,  $H_{j,g_B}$ , e a taxa de emissão de pulsos,  $\omega_{j,g_B}$ , devem inicializadas com valores distintos para cada morcego e sua escolha pode ser feita de maneira randômica. Valores típicos para a inicialização de  $H_{j,g_B}$  encontram-se no intervalo  $[1,2]$ , enquanto os valores recomendados para a inicialização de  $\omega_{j,g_B}$  estão no intervalo  $[0,1]$  (YANG, 2010).

A amplitude dos pulsos,  $H_{j,g_B}$ , e a taxa de emissão de pulsos,  $\omega_{j,g_B}$ , podem ser atualizadas a cada iteração, conforme as equações a seguir:

$$H_{j,g_B+1} = \eta H_{j,g_B} \quad (3.16)$$

$$\omega_{j,g_B+1} = \omega_{j,0}[1 - \exp(-\vartheta g_B)] \quad (3.17)$$

onde  $\eta$  e  $\vartheta$  são constantes, sendo possível adotar  $\eta = \vartheta$  para uma execução simplificada do algoritmo.

Os valores de  $H_{j,g_B}$  e de  $\omega_{j,g_B}$  de um determinado morcego só deverão ser atualizados caso ocorra uma melhoria na sua posição após a busca local, ou seja, se  $J_B(\mathbf{b}_{j,g_B}^*) < J_B(\mathbf{b}_{j,g_B})$ , o que significa que aquele morcego está se movimentando em direção à solução ótima.

Na literatura é possível encontrar estudos que apresentam versões melhoradas do BA (JAMIL; ZEPERNIC; YANG, 2013; KHAN; NIKOV; SAHAI, 2011; LIN et al., 2012; XIE; ZHOU; CHEN, 2013; YANG, 2011). Entretanto, neste trabalho será aplicada a versão básica que foi detalhada nesta seção.

### 3.5 RESUMO DO CAPÍTULO

Este capítulo abordou a definição de metaheurísticas, além de contextualizar sua aplicabilidade em problemas de otimização, como por exemplo, a seleção de estrutura na área de identificação de sistemas. Foram apresentadas em detalhes as versões básicas das metaheurísticas que serão aplicadas neste trabalho, bem como referências que tratam de versões melhoradas das mesmas. Conceitos relacionados ao Algoritmo Genético foram explorados e seus operadores de seleção natural, seleção para reprodução, cruzamento e mutação foram apresentados. As estratégias básicas da Evolução Diferencial foram abordadas e os operadores de mutação, cruzamento e seleção foram detalhados matematicamente. Além disso, foram apresentados os principais conceitos do Algoritmo do Morcego, bem como as regras básicas que regem seu funcionamento.

## 4 CODIFICAÇÃO BINÁRIA PARA METAHEURÍSTICAS

### 4.1 INTRODUÇÃO

Os problemas de otimização podem ser divididos em dois tipos: problemas baseados em variáveis inteiras ou discretas e problemas baseados em variáveis contínuas (SIARRY, 2016). Exemplos típicos de problemas com variáveis discretas são variantes de problemas de atribuição e programação, localização, layout de instalações, particionamento e cobertura de conjuntos, controle de estoque, roteamento de veículos, caixeiro viajante, entre outros (GOLDBARG; GOLDBARG; LUNA, 2015; GRAHAM, 1995).

Dentro dos problemas discretos, também chamados de combinatoriais, existe uma categoria chamada de problemas de otimização binária, onde as variáveis do problema só podem assumir os valores 0 ou 1. Um exemplo típico deste tipo de problema na área de identificação de sistemas é a seleção de estrutura, conforme visto na seção 2.4.

Neste contexto, metaheurísticas vêm sendo cada vez mais aplicadas para resolver problemas de otimização binária. Algumas delas foram originalmente criadas para lidar com problemas binários, enquanto outras foram projetadas para trabalhar com variáveis contínuas e necessitam de alguma adaptação para tratar deste tipo de problema (DAHI; MEZIOUD; DRAA, 2015).

As técnicas de codificação binária que normalmente são aplicadas em metaheurísticas projetadas para espaço contínuo encontram-se divididas em dois grupos principais. O primeiro grupo contém técnicas conhecidas como codificação binária em dois passos. Com estas técnicas é possível trabalhar com as metaheurísticas de espaço contínuo sem que seus operadores sejam modificados. Neste caso, a codificação binária é realizada através de duas etapas aplicadas após as iterações normais das metaheurísticas, transformando, desta maneira, as soluções contínuas em binárias. No segundo grupo, estão presentes técnicas chamadas de transformação de operadores contínuos em binários. Ao utilizar este tipo de técnica, são feitas modificações diretamente nos operadores das metaheurísticas, permitindo que estas trabalhem com soluções binárias (CRAWFORD et al., 2017).

Nesta pesquisa, metaheurísticas voltadas para o espaço de busca contínuo terão suas versões binárias implementadas através de algumas técnicas de

codificação binária em dois passos, as quais serão apresentadas nas seções seguintes.

## 4.2 CODIFICAÇÃO BINÁRIA POR FUNÇÃO DE TRANSFERÊNCIA

Esta técnica é aplicada em duas etapas: inicialmente é feita uma normalização através de uma função de transferência (TF, do inglês *Transfer Function*) e, em seguida, as soluções são convertidas para um formato binário através de uma regra de codificação binária. Este método foi aplicado pela primeira vez à otimização por enxame de partículas, onde a TF era responsável por fornecer valores de probabilidade e buscou modelar a transição das posições das partículas (KENNEDY; EBERHART, 1997). Um tipo de função utilizada nesta técnica (CRAWFORD et al., 2017) é dado pela equação (4.1), sendo  $\mu$  o fator utilizado para ajustar a forma da função:

$$T(\alpha_i) = \frac{1}{1 + e^{-\mu\alpha_i}} \quad (4.1)$$

Considerando  $A = \{\alpha_1, \dots, \alpha_n\}$  como uma solução encontrada pela metaheurística que trabalha com variáveis contínuas, em cada dimensão é aplicada a função de transferência, obtendo  $\gamma_i = T(\alpha_i)$ , de tal forma que  $\Gamma = \{\gamma_1, \dots, \gamma_n\}$  torna-se uma solução intermediária, onde  $\gamma_i \in [0,1]$ .

Com a solução intermediária determinada, deve-se aplicar alguma regra de codificação binária para se obter a solução  $B = \{\beta_1, \dots, \beta_n\}$ , a qual estará em formato binário, ou seja,  $\beta_i \in \{0,1\}$ . Alguns exemplos de regras aplicadas nesta etapa são:

$$\beta_i = \begin{cases} 1, & \text{se } h \leq \gamma_i \\ 0, & \text{se } h > \gamma_i \end{cases} \quad (4.2)$$

$$\beta_i = \begin{cases} \text{complemento}(\beta_{i-1}), & \text{se } h \leq \gamma_i \\ 0, & \text{se } h > \gamma_i \end{cases} \quad (4.3)$$

Em todos os casos,  $h$  é um número que deve ser escolhido no intervalo  $[0,1]$ .

### 4.3 CODIFICAÇÃO BINÁRIA POR PRIORIDADE DO MAIOR VALOR

A técnica de prioridade do maior valor (GVP, do inglês *Great Value Priority*) foi inicialmente aplicada para resolver um problema de atribuição quadrática utilizando otimização por enxame de partículas (CONGYING; HUANPING; XINFENG, 2011). Este método codifica um espaço contínuo  $\mathbb{R}^n$  em um espaço binário  $\{0,1\}^n$  através de duas principais propriedades: consiste em um mapeamento injetivo e reflete uma relação de ordem prioritária, adequada para problemas de atribuição.

Considerando uma possível solução  $A = \{\alpha_1, \dots, \alpha_n\}$ , encontrada pela metaheurística, o primeiro passo é obter o vetor de posições  $\Lambda = \{\lambda_1, \dots, \lambda_n\}$ , onde  $\Lambda \in \mathbb{Z}^n$ , relacionado aos elementos de  $A$ . A regra GVP faz um ordenamento dos elementos de  $A$ , organizando-os do maior para o menor e identificando sua posição nesta lista ordenada. Em seguida, estas novas posições são indicadas em  $\Lambda$ , obedecendo os índices das antigas posições de cada elemento. Por fim, obtém-se o vetor  $B = \{\beta_1, \dots, \beta_n\}$ ,  $\beta_i \in \{0,1\}$ , aplicando-se a regra apresentada na equação (4.7).

$$\beta_i = \begin{cases} 1, & \text{se } \lambda_i > \lambda_{i+1} \\ 0, & \text{se } \lambda_i < \lambda_{i+1} \end{cases} \quad (4.4)$$

A TABELA 2 apresenta um exemplo de codificação binária utilizando o método GVP.

TABELA 2 – Exemplo de aplicação da codificação binária utilizando GVP

A	5,7	6,5	6,1	8,3	7,9	4,2
$\Lambda$	5	3	4	1	2	6
B	1	0	1	0	0	1

FONTE: O autor (2020).

### 4.4 CODIFICAÇÃO BINÁRIA POR MODULAÇÃO EM ÂNGULO

A técnica de modulação em ângulo (AM, do inglês *Angle Modulation*) é derivada da área de telecomunicações e é aplicada em processamento de sinais

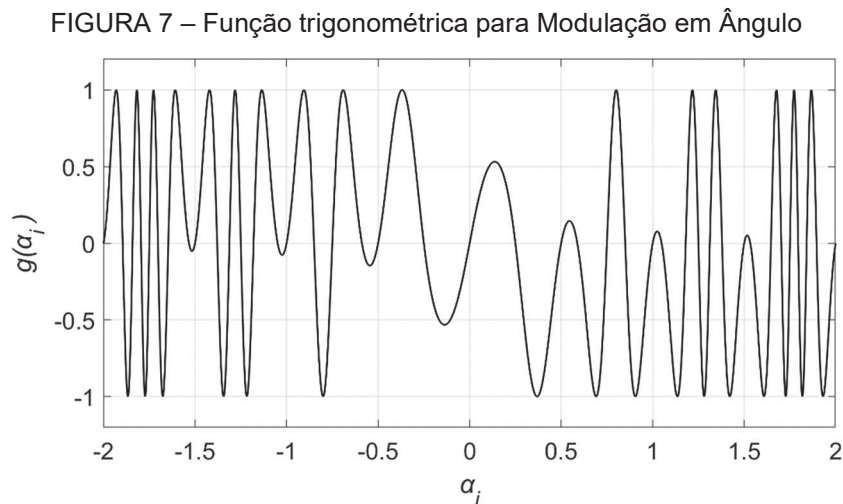


(PROAKIS et al., 2002). Em otimização heurística binária, foi inicialmente utilizada em conjunto com a otimização por enxame de partículas aplicada a algumas funções de *benchmark* (PAMPARA; FRANKEN; ENGELBRECHT, 2005). Neste método, a codificação binária é feita a partir de um vetor contínuo  $A = \{\alpha_1, \dots, \alpha_n\}$ , o qual terá a dimensão  $n$  do problema e será formado por valores fixos e uniformemente espaçados em um determinado intervalo. Para obter a solução binária  $B = \{\beta_{1j}, \beta_{2j}, \dots, \beta_{nj}\}$ , utilizam-se a função trigonométrica dada pela equação (4.5) e a regra da equação (4.6):

$$g_j(\alpha_i) = \sin\left(2\pi \cdot (\alpha_i - \rho_{1j}) \cdot \rho_{2j} \cdot \cos\left(2\pi \cdot (\alpha_i - \rho_{1j}) \cdot \rho_{3j}\right)\right) + \rho_{4j} \quad (4.5)$$

$$\beta_{ij} = \begin{cases} 1, & \text{se } g_j(\alpha_i) \geq 0 \\ 0, & \text{se } g_j(\alpha_i) < 0 \end{cases} \quad (4.6)$$

Os coeficientes da equação (4.5) determinam a forma da função geradora. O coeficiente  $\rho_{1j}$  representa o deslocamento horizontal da função, os parâmetros  $\rho_{2j}$  e  $\rho_{3j}$  representam as frequências das funções seno e cosseno, respectivamente, e o coeficiente  $\rho_{4j}$  determina o deslocamento vertical da função. Um exemplo desta função, com  $\alpha_i \in [-2, 2]$ ,  $\rho_{1j} = \rho_{4j} = 0$  e  $\rho_{2j} = \rho_{3j} = 1$ , é mostrado na FIGURA 7.



FONTE: O autor (2020).

O primeiro passo na aplicação da codificação binária AM é utilizar um espaço de busca que possui dimensão igual a 4, onde cada dimensão representa um coeficiente da equação (4.5). Neste espaço de busca, cada solução do tipo  $\boldsymbol{\rho}_j = \{\rho_{1j}, \rho_{2j}, \rho_{3j}, \rho_{4j}\}$ , localizada pela metaheurística, gera uma função trigonométrica  $g_j$  diferente. Desta forma, para cada solução  $\boldsymbol{\rho}_j = \{\rho_{1j}, \rho_{2j}, \rho_{3j}, \rho_{4j}\}$  no espaço de 4 dimensões, é gerada uma solução binária  $B = \{\beta_{1j}, \beta_{2j}, \dots, \beta_{nj}\}$  de dimensão  $n$ . Ou seja, a busca é feita um espaço de dimensão 4, independente de qual seja a dimensão  $n$  do problema.

#### 4.5 CODIFICAÇÃO BINÁRIA POR MODULAÇÃO EM ÂNGULO MODIFICADA

Além de utilizar a codificação binária AM em sua versão usual, como foi apresentada na seção 4.4, este trabalho propõe uma modificação nesta técnica. Nesta modificação, em vez de fixar o vetor  $A = \{\alpha_1, \dots, \alpha_n\}$  com números uniformemente espaçados em um determinado intervalo, serão fixados os coeficientes  $\rho_{1j}$ ,  $\rho_{2j}$ ,  $\rho_{3j}$  e  $\rho_{4j}$  da função geradora  $g_j$ , e a metaheurística atuará na busca do vetor  $A$ , que será codificado em uma solução binária  $B$  utilizando as equações (4.5) e (4.6). Neste caso, o espaço de busca deverá manter a dimensão  $n$  do problema. Para fins de notação, esta técnica será chamada de modulação em ângulo modificada (MAM, do inglês *Modified Angle Modulation*).

#### 4.6 RESUMO DO CAPÍTULO

Neste capítulo foram abordados conceitos que possibilitam metaheurísticas originalmente projetadas para espaços contínuos atuarem em um espaço de busca binário. Foram tratadas algumas técnicas de codificação binária em dois passos, que serão aplicadas neste trabalho. Apresentou-se a teoria sobre as codificações TF, GVP e AM, explicando como estas técnicas adaptam metaheurísticas voltadas para espaço contínuo para trabalharem com soluções binárias. Além disso, foi apresentada uma

proposta de modificação na codificação por modulação em ângulo (MAM), que será objeto de investigação nesta pesquisa.

## 5 METODOLOGIA

### 5.1 INTRODUÇÃO

A fundamentação teórica apresentada nos capítulos anteriores mostrou os principais conhecimentos necessários ao desenvolvimento da investigação proposta neste trabalho. De posse das teorias apresentadas, é possível implementar o algoritmo que será aplicado neste trabalho para seleção de estrutura. De modo geral, este algoritmo é apresentado no fluxograma da FIGURA 8.

Nas seções seguintes, serão dados detalhes a respeito de cada etapa do algoritmo e sobre a avaliação de desempenho das técnicas utilizadas.

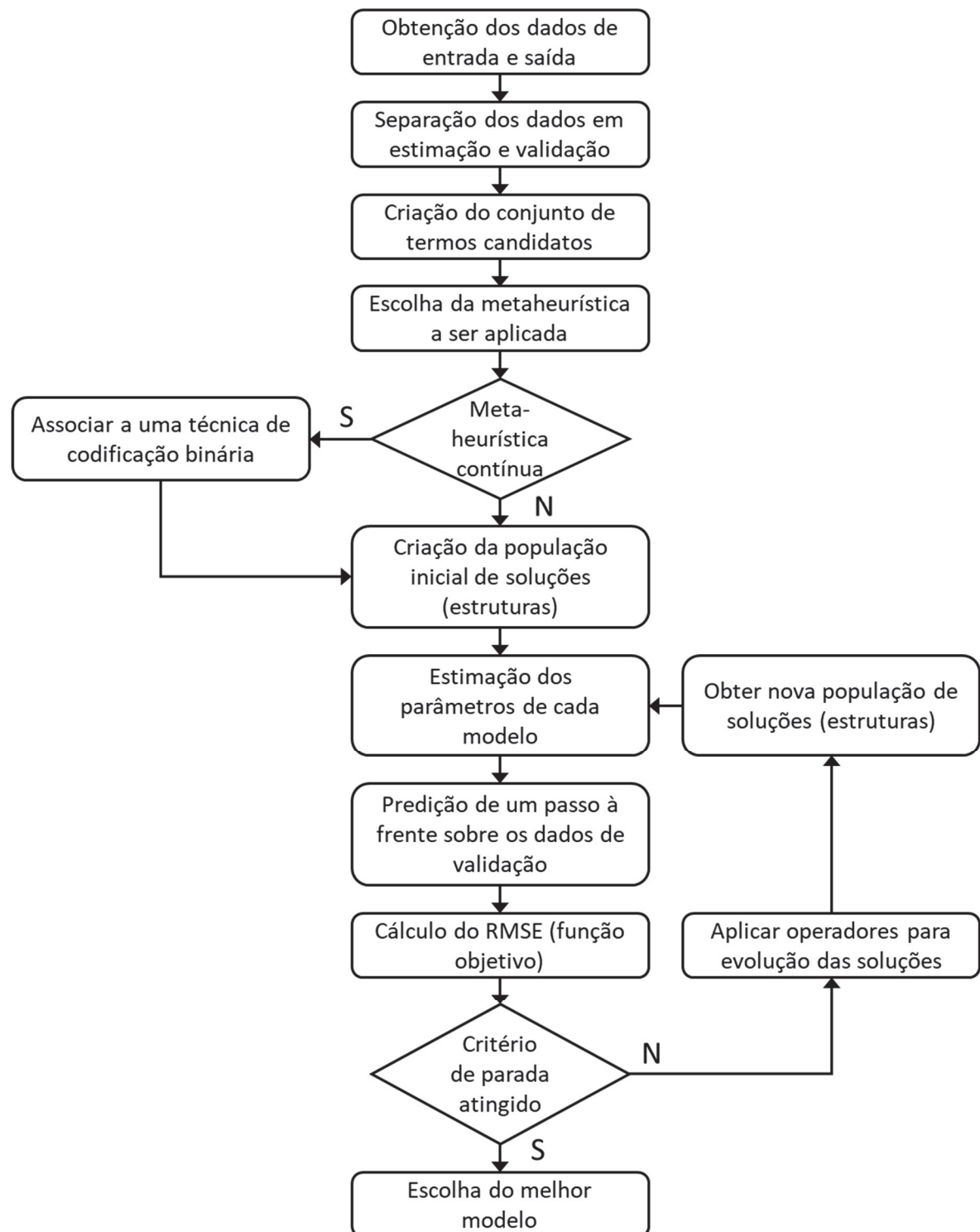
### 5.2 OBTENÇÃO DOS DADOS E SEPARAÇÃO EM DOIS CONJUNTOS

Inicialmente, deve-se dispor de dados experimentais obtidos dos sistemas para os quais serão gerados os modelos. Estes dados devem ser separados em dois conjuntos distintos: um para construção dos modelos, e o outro para validação dos modelos. Nesta etapa, serão utilizados dados disponíveis na literatura, oriundos de estudos realizados por outros pesquisadores.

### 5.3 CRIAÇÃO DO CONJUNTO DE TERMOS CANDIDATOS

A etapa seguinte, é a criação do conjunto de termos candidatos, que formará o espaço de busca das metaheurísticas. Para criar este conjunto, é necessário selecionar um tipo de modelo e escolher os valores de atrasos e grau de não linearidade. Neste trabalho, o tipo de modelo utilizado será o NARX, com o parâmetro  $d$  igual a 1 em todos os casos. Desta forma, devem ser escolhidos os valores de  $l$ ,  $n_y$  e  $n_u$  e os termos candidatos podem ser determinados através das equações (2.5) e (2.6).

FIGURA 8 – Fluxograma do procedimento aplicado para seleção de estrutura e obtenção de modelos.

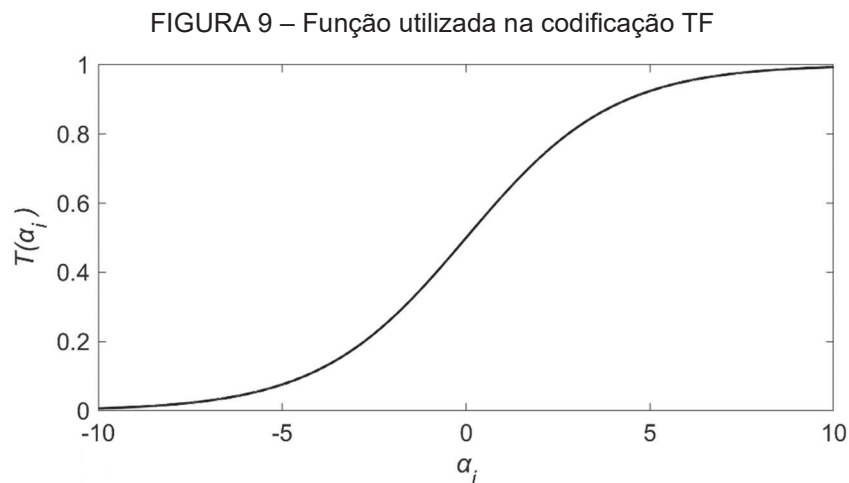


FONTE: O autor (2020).

## 5.4 ESCOLHA DA METAHEURÍSTICA

Em seguida deve-se determinar qual metaheurística será utilizada, a qual irá manipular soluções do tipo binário para realizar a seleção de estrutura. No caso do GA, as soluções originalmente já são tratadas em formato binário. Contudo, para a DE e o BA, será necessário aplicar codificações binárias para que as soluções sejam obtidas no formato desejado. No capítulo 4 foram apresentadas as seguintes codificações binárias para metaheurísticas que atuam em espaço contínuo: TF, GVP, AM e MAM.

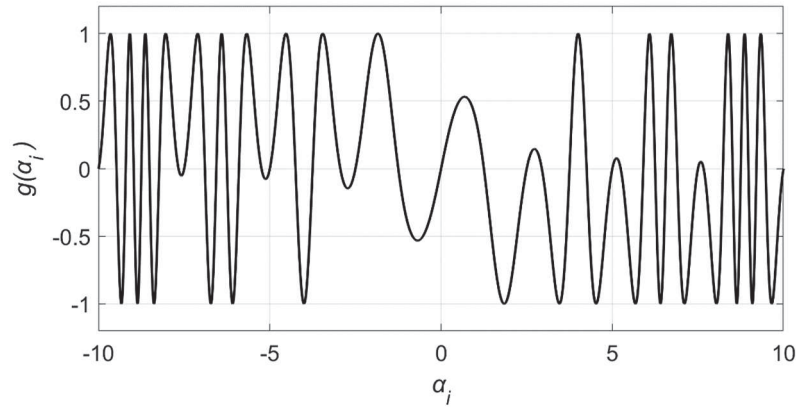
Para a codificação TF, neste trabalho será aplicada a função dada pela equação (4.1) com  $\mu = 1/2$ , cujo gráfico pode ser observado na FIGURA 9. A regra de codificação aplicada será a da equação (4.2) e será adotado  $h = 0,7$ .



FONTE: O autor (2020).

Para a codificação GVP, não há parâmetros a serem ajustados. A codificação AM, por sua vez, utilizará o vetor  $A = \{\alpha_1, \dots, \alpha_n\}$ , com números uniformemente espaçados no intervalo  $[-10, 10]$ . Já a codificação MAM terá como coeficientes para a função da equação (4.5) os valores  $\rho_1 = \rho_4 = 0$  e  $\rho_2 = \rho_3 = 0,2$ . Conforme apresentado na seção 4.5, a codificação MAM utilizará uma função geradora única, a qual pode ser observada na FIGURA 10.

FIGURA 10 – Função geradora utilizada na codificação MAM



FONTE: O autor (2020).

### 5.5 POPULAÇÃO INICIAL E AVALIAÇÃO DA APTIDÃO DE SOLUÇÕES

A etapa seguinte consiste na criação de uma população inicial de soluções e no cálculo da aptidão de cada uma delas através da função objetivo. A população inicial será criada de maneira aleatória a partir de uma distribuição uniforme, mantendo-se a mesma semente de geração de números randômicos para todas as técnicas.

A avaliação da aptidão das soluções será realizada da seguinte maneira: considerando, por exemplo,  $l = 2$ ,  $n_y = 1$  e  $n_u = 1$ , o conjunto de termos candidatos seria:

$$R_\theta = \{1, y(k-1), u(k-1), y^2(k-1), y(k-1)u(k-1), u^2(k-1)\} \quad (5.1)$$

onde o número 1 aparece para possibilitar o cálculo da constante  $\theta_0$ . Caso a metaheurística apresente a seguinte solução  $B = \{1, 1, 0, 0, 0, 1\}$ , significa que somente o primeiro, o segundo e o sexto termos farão parte do modelo, que terá o seguinte formato:

$$y(k) = \theta_0 + \theta_1 y(k-1) + \theta_2 u^2(k-1) + e(k) \quad (5.2)$$

O passo seguinte consiste em estimar os parâmetros do modelo através do algoritmo MQ e obter os valores de  $\theta_0$ ,  $\theta_1$  e  $\theta_2$ .

O desempenho de uma solução  $B_j$  será baseado na validação do modelo gerado por esta solução. Nesta etapa, a validação utilizada será a predição de um passo à frente e o índice aplicado será o *RMSE*. Desta forma, a função objetivo que deverá ser minimizada, direcionando a busca da metaheurística, será:

$$J(B_j) = \sqrt{\frac{\sum_{k=1}^N (y(k) - \hat{y}(k))^2}{N}} \quad (5.3)$$

onde  $y(k)$  e  $\hat{y}(k)$  são, respectivamente, as saídas observada e predita utilizando os dados de validação, e  $N$  é a quantidade de dados utilizados.

## 5.6 EVOLUÇÃO DA POPULAÇÃO DE SOLUÇÕES

A partir deste ponto, inicia-se o ciclo iterativo da metaheurística, onde são aplicados os operadores para evoluir a população de soluções. A cada vez que este ciclo é executado, conta-se uma geração e o mesmo se repetirá até que algum critério de parada seja atingido. Neste trabalho, o critério de parada utilizado será a quantidade máxima de gerações.

Uma vez atingido o critério de parada, o algoritmo encerrará a busca e o melhor modelo obtido será aquele gerado pela solução  $B_j$  que apresentar o menor valor de  $J(B_j)$ . Com a finalização desta etapa e seleção do melhor modelo, tem-se a conclusão de uma simulação. Para fins de análise do desempenho das técnicas apresentadas, cada simulação será repetida várias vezes.

## 5.7 AVALIAÇÃO DO DESEMPENHO DAS TÉCNICAS

Algumas metaheurísticas serão avaliadas em mais de uma configuração. Para o GA, serão avaliadas as seleções do tipo roleta e torneio, bem como os cruzamentos de um único ponto e de dois pontos. Por questões de notação, estas abordagens serão apresentadas conforme indicado na TABELA 3.



TABELA 3 – Abordagens utilizadas pelo Algoritmo Genético

<b>Seleção para Reprodução</b>	<b>Cruzamento</b>	<b>Notação</b>
Roleta	Um ponto	GA/Roleta/1P
Roleta	Dois pontos	GA/Roleta/2P
Torneio	Um ponto	GA/Torneio/1P
Torneio	Dois pontos	GA/Torneio/2P

FONTE: O autor (2020).

Para a DE, serão verificadas as quatro primeiras estratégias apresentadas na TABELA 1, ou seja: DE/rand/1/bin, DE/rand/2/bin, DE/best/1/bin e DE/best/2/bin. O BA, por sua vez, será utilizado em uma única configuração. Além disso, as versões binárias da DE e do BA serão implementadas utilizando as codificações TF, GVP, AM e MAM. Desta forma, tem-se um total de 24 técnicas a serem avaliadas no processo de seleção de estrutura. Estas técnicas encontram-se listadas na TABELA 4.

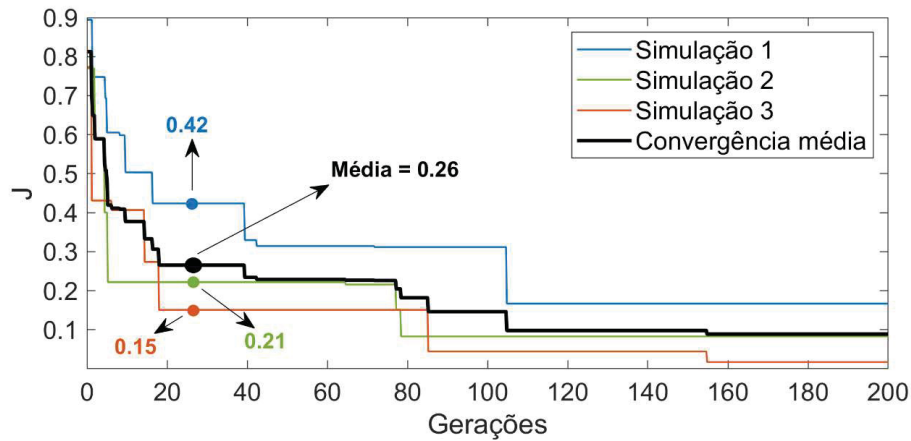
TABELA 4 – Técnicas aplicadas no algoritmo de seleção de estrutura

<b>Codificação Binária</b>	<b>Estratégia</b>
-	GA/Roleta/1P
-	GA/Roleta/2P
-	GA/Torneio/1P
-	GA/Torneio/2P
TF	DE/rand/1/bin
	DE/rand/2/bin
	DE/best/1/bin
	DE/best/2/bin
	BA
GVP	DE/rand/1/bin
	DE/rand/2/bin
	DE/best/1/bin
	DE/best/2/bin
	BA
AM	DE/rand/1/bin
	DE/rand/2/bin
	DE/best/1/bin
	DE/best/2/bin
	BA
MAM	DE/rand/1/bin
	DE/rand/2/bin
	DE/best/1/bin
	DE/best/2/bin
	BA

FONTE: O autor (2020).

Para avaliar o desempenho de cada técnica, inicialmente será verificado o comportamento médio da curva de convergência da função objetivo. Para isto, em cada geração serão tomados os valores obtidos por todas as simulações e tirada a média. Um exemplo desta análise, para um caso fictício onde foram realizadas 3 simulações, pode ser observado na FIGURA 11.

FIGURA 11 – Exemplo de curva de convergência média



FONTE: O autor (2020).

Outra maneira de avaliar o desempenho das técnicas será através do valor da função objetivo alcançado ao final de cada simulação. Neste caso, serão verificados os valores mínimo, médio, máximo e desvio padrão dentre todas as simulações realizadas por cada técnica.

Considerando, ainda, que o objetivo do algoritmo é localizar um modelo classificado como ótimo, ou seja, com o menor *RMSE* possível, será adotado um índice,  $L$ , para medir a taxa de sucesso de cada estratégia em encontrar um determinado modelo de referência. Este índice será dado por:

$$L = \frac{S_{\alpha}}{S_t} \times 100\% \quad (5.4)$$

onde  $L$  é a taxa de localização,  $S_{\alpha}$  é quantidade de simulações em que o modelo de referência foi localizado e  $S_t$  é o total de simulações realizadas pela técnica em questão.

As técnicas também serão comparadas quanto ao custo computacional exigido durante a execução do algoritmo. Para isto será mensurado o tempo de realização de cada uma das  $S_t$  simulações e posteriormente serão verificados os valores de média e desvio padrão.

Além de analisar o desempenho das técnicas aplicadas na seleção de estrutura, os melhores modelos obtidos serão avaliados por meio dos métodos de validação apresentados na seção 2.6. Como o processo de busca das metaheurísticas já é direcionado por uma validação baseada em predição de um passo à frente, os desempenhos dos modelos selecionados ao final das simulações serão avaliados através de simulação livre e análise de resíduos, além de serem feitas comparações com modelos apresentados na literatura.

Todos os códigos utilizados neste trabalho foram implementados em ambiente MATLAB™ e as simulações foram executadas em um computador desktop com processador Intel Core i7 e 16 GB de memória RAM (do inglês *Random Access Memory*).

## 5.8 RESUMO DO CAPÍTULO

Neste capítulo foi apresentada a metodologia que será aplicada neste trabalho, sendo detalhada cada etapa que a compõe. Foram listadas as abordagens do GA e as estratégias da DE que foram selecionadas, além do BA, que serão empregados no algoritmo de seleção de estrutura, perfazendo um total de 24 configurações a serem analisadas. Foram apresentadas as configurações para as codificações binárias que serão associadas à DE e ao BA. Comentou-se também acerca dos indicadores que serão utilizados para análise de desempenho das técnicas aplicadas e para validação dos modelos obtidos.

## 6 ESTUDOS DE CASO

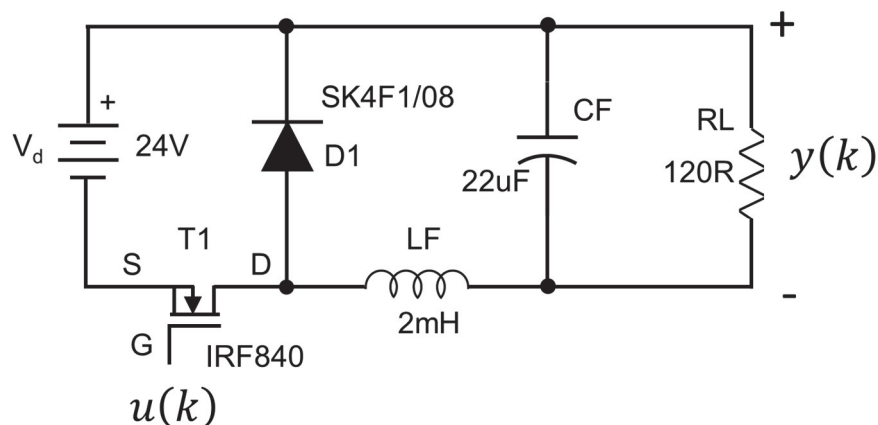
### 6.1 INTRODUÇÃO

Este capítulo tem como objetivo apresentar a aplicação da metodologia tratada no capítulo 5. Para isto, serão utilizados estudos de caso, onde as metaheurísticas GA, DE e BA atuarão na seleção de estrutura de modelos polinomiais NARX para os sistemas apresentados.

### 6.2 ESTUDO DE CASO 1: CONVERSOR *BUCK*

Nesta seção será apresentado o conversor *buck* utilizado neste estudo de caso. Este conversor foi discutido inicialmente no trabalho de Aguirre, Donoso-Garcia e Santos-Filho (2000) e vem sendo frequentemente utilizado como sistema de teste na área de identificação de sistemas, tanto para procedimentos do tipo caixa-preta, quanto caixa-cinza (BARROSO; NEPOMUCENO, 2004; CORRÊA; AGUIRRE; SALDANHA, 2002; MARTINS; NEPOMUCENO; BARROSO, 2013; SEVERINO; ARAÚJO, 2017). O diagrama do conversor *buck* é apresentado na FIGURA 12.

FIGURA 12 – Diagrama do conversor *buck*



FONTE: (AGUIRRE; DONOSO-GARCIA; SANTOS-FILHO, 2000).

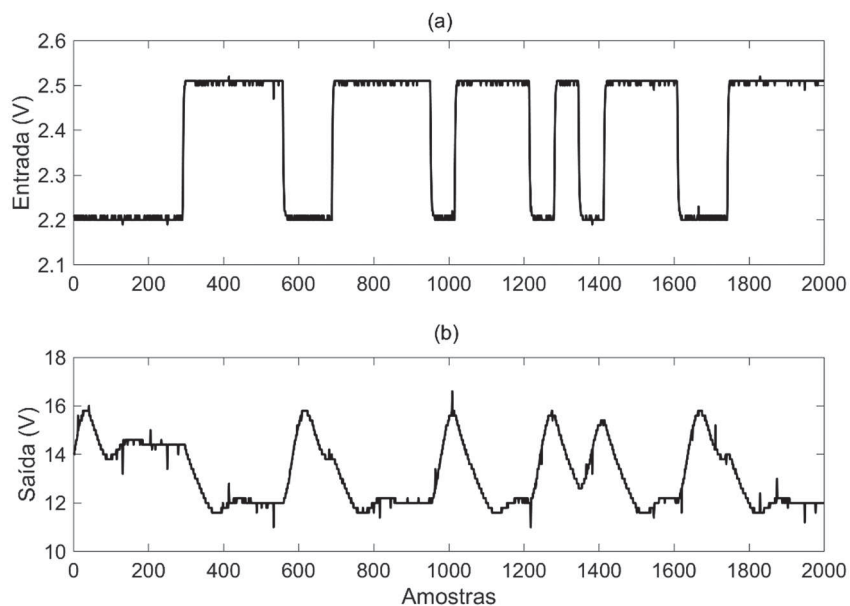
O sistema de regulação de tensão de carga não é mostrado na FIGURA 12. No experimento realizado com este conversor, a fonte de alimentação  $v_d$  foi mantida constante e igual a 24V. Para excitar a dinâmica do conversor, optou-se por um sinal

de entrada do tipo PRBS, limitado entre  $2,2V$  e  $2,5V$ . O MOSFET IRF840 foi chaveado atuando-se na porta G. A razão cíclica é definida como a proporção de tempo em que a chave está ligada em relação ao período de operação,  $T_t$ , ou seja,  $D = T_{ligado}/T_t$ . A razão cíclica foi variada usando-se técnicas de modulação por largura de pulso (PWM, do inglês *Pulse Width Modulation*) a uma taxa de  $1/T_t = 33kHz$ , por meio do circuito integrado LM3524. Para excitar a dinâmica do conversor, optou-se por um sinal de entrada do tipo PRBS, aplicado ao *buck* usando-se um conversor Digital/Analógico (AGUIRRE; DONOSO-GARCIA; SANTOS-FILHO, 2000).

Testes preliminares, que consistiram basicamente de respostas ao degrau, mostraram que o conversor tem uma constante de tempo dominante de aproximadamente  $2ms$ . Contudo, o tempo de amostragem escolhido foi  $T_s = 10\mu s$ . Este valor é menor do que o necessário, mas preferiu-se superamostrar os dados para testar diferentes fatores de decimação e, conseqüentemente, diferentes taxas de amostragem efetiva (AGUIRRE; DONOSO-GARCIA; SANTOS-FILHO, 2000).

Neste experimento, a tensão que define a razão cíclica do conversor foi considerada como sinal de entrada,  $u(k)$ , e a tensão elétrica na saída do conversor foi adotada como sinal de saída,  $y(k)$ . Foram coletados 2000 pares de amostras, que podem ser observados na FIGURA 13.

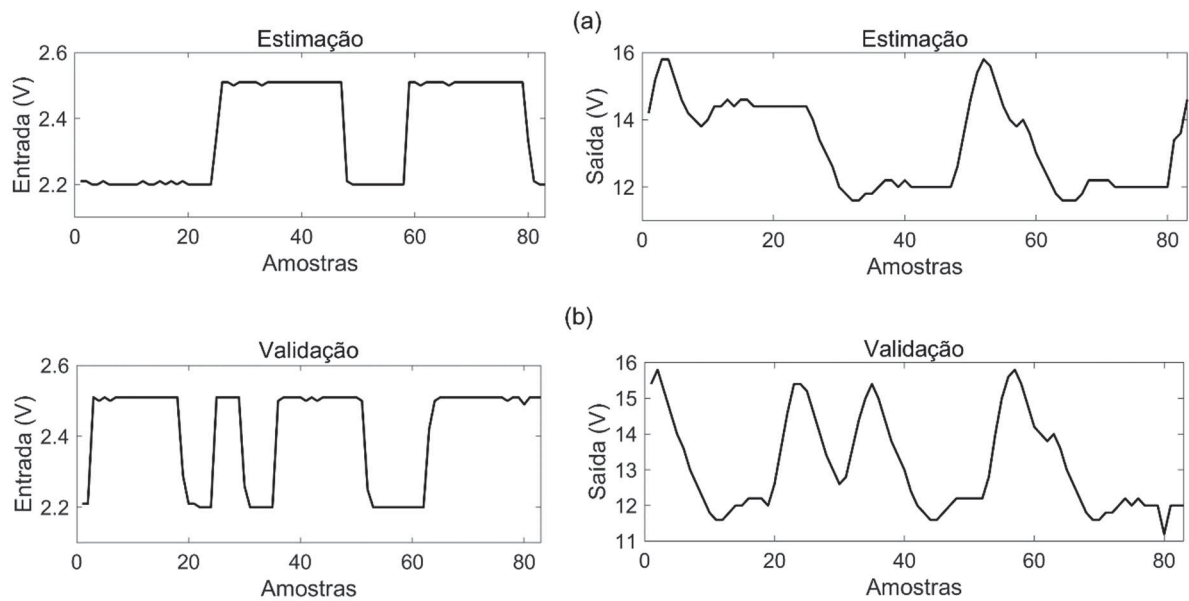
FIGURA 13 – Dados coletados do conversor *buck*. (a) sinal de entrada e (b) sinal de saída.



FONTE: (AGUIRRE; DONOSO-GARCIA; SANTOS-FILHO, 2000).

Como os dados foram amostrados a uma taxa mais alta do que a necessária, optou-se por aplicar um tempo de amostragem de trabalho igual a  $T_s^* = 120\mu s$ , resultando num total de 166 pares de amostras. Além disso, os dados foram divididos em dois conjuntos distintos, cada um com 50% dos 166 pares de amostras. A primeira metade dos dados foi utilizada para estimação dos modelos e a segunda metade foi reservada para a etapa de validação, conforme observado na FIGURA 14.

FIGURA 14 – Sinais de entrada e saída do conversor *buck*. Dados de estimação (a) e validação (b).



FONTE: O autor (2020).

Os dados apresentados na FIGURA 14 foram utilizados por Aguirre, Donoso-Garcia e Santos-Filho (2000) para obtenção de modelos para o conversor *buck*. Estes mesmos dados foram disponibilizados na internet pelos autores, o que permitiu sua utilização neste estudo de caso.

### 6.2.1 Parâmetros do algoritmo

Para aplicar o algoritmo apresentado no capítulo 5, inicialmente foi necessário determinar alguns parâmetros relacionados ao conjunto de termos candidatos e às metaheurísticas.

O conjunto de termos candidatos foi gerado considerando  $n_y = 3$ ,  $n_u = 3$ ,  $l = 2$  e  $d = 1$ , seguindo o mesmo procedimento adotado por Aguirre, Donoso-Garcia e Santos-Filho (2000). Foram selecionados os mesmos valores apresentados pelos autores para que a comparação entre o modelo por eles apresentado e os modelos obtidos neste trabalho ocorresse de forma mais coerente. Os termos possíveis para compor os modelos estão apresentados na FIGURA 15.

FIGURA 15 – Conjunto de termos candidatos para geração de modelos do conversor *buck*.

1º	1	8º	$y^2(k-1)$	15º	$y^2(k-2)$	22º	$y(k-3)u(k-3)$
2º	$y(k-1)$	9º	$y(k-1)y(k-2)$	16º	$y(k-2)u(k-1)$	23º	$u^2(k-1)$
3º	$y(k-2)$	10º	$y(k-1)y(k-3)$	17º	$y(k-2)u(k-2)$	24º	$u(k-1)u(k-2)$
4º	$y(k-3)$	11º	$y(k-1)u(k-1)$	18º	$y(k-2)u(k-3)$	25º	$u(k-1)u(k-3)$
5º	$u(k-1)$	12º	$y(k-1)u(k-2)$	19º	$y^2(k-3)$	26º	$u^2(k-2)$
6º	$u(k-2)$	13º	$y(k-1)u(k-3)$	20º	$y(k-3)u(k-1)$	27º	$u(k-2)u(k-3)$
7º	$u(k-3)$	14º	$y(k-2)y(k-3)$	21º	$y(k-3)u(k-2)$	28º	$u^2(k-3)$

FONTE: O autor (2020).

A quantidade de termos candidatos é o que irá determinar o tamanho dos vetores binários manipulados pelas metaheurísticas. Ou seja, neste caso tem-se  $N_{V_G} = N_{V_D} = N_{V_B} = 28$ . Para esta quantidade de termos candidatos, existem  $2^{28} - 1 = 268.435.455$  estruturas possíveis de serem utilizadas. Para o tamanho das populações das metaheurísticas foi adotado o valor  $N_{P_G} = N_{P_D} = N_{P_B} = 100$ . O número máximo de gerações adotado foi  $g_{G_{max}} = g_{D_{max}} = g_{B_{max}} = 1000$ , considerado como critério de parada do algoritmo.

Valores de parâmetros específicos de cada metaheurística foram selecionados heurísticamente através de testes independentes. Nestes testes foram realizadas simulações onde os parâmetros já citados até aqui foram fixados nos seus valores indicados e os demais parâmetros a serem determinados eram variados dentro de uma determinada faixa. Para cada combinação de parâmetros a determinar, foram realizadas 10 repetições da simulação e, ao final, foi tomada a média dos valores resultantes da função objetivo em cada repetição. Os conjuntos de parâmetros que apresentaram as menores médias como resultado foram considerados os melhores, sendo adotados para o estudo de caso.

Para o GA, os parâmetros a serem determinados eram: a taxa de seleção natural,  $t_{sn}$ , testada no intervalo  $[0,1, 0,9]$ ; a taxa de cruzamento,  $t_c$ , testada no intervalo  $[0,6, 0,9]$ ; e a probabilidade de mutação,  $p_m$  testada no intervalo  $[0,001, 0,05]$ . Os intervalos de teste foram escolhidos conforme indicações da literatura (BEASLEY; BULL; MARTIN, 1993; SRINIVAS; PATNAIK, 1994; HAUPT; HAUPT, 2004). O conjunto de parâmetros que apresentou o menor valor médio para a função objetivo, após as 10 repetições da simulação, está apresentado na TABELA 5.

Para a DE, os parâmetros a serem determinados eram: o fator de escala,  $F_m$ , testado no intervalo  $[0, 2]$ ; e a probabilidade de cruzamento,  $C_r$ , testada no intervalo  $[0, 1]$ . Estes intervalos também foram selecionados conforme indicações da literatura (PRICE, 1996; STORN; PRICE, 1997; PRICE; STORN; LAMPINEN, 2005). Os valores dos parâmetros que possibilitaram o melhor desempenho do algoritmo estão apresentados na TABELA 5.

No caso do BA, os parâmetros a serem determinados eram: a constante de atualização de amplitudes de pulsos,  $\eta$ , testada no intervalo  $[0, 1]$ ; e a constante de atualização da taxa de emissão de pulsos  $\vartheta$ , também testada no intervalo  $[0, 1]$ . Estes intervalos foram selecionados conforme indicações da literatura (YANG, 2010; YANG; GANDOMI, 2012). Os melhores valores encontrados para estes parâmetros estão apresentados na TABELA 5.

TABELA 5 – Parâmetros das metaheurísticas (Estudo de caso 1)

Metaheurística	Parâmetro	Valor
GA	$t_{sn}$	0,5
	$t_c$	0,7
	$p_m$	0,05
DE	$F_m$	0,8
	$C_r$	0,7
BA	$\eta$	0,7
	$\vartheta$	0,1

FONTE: O autor (2020).

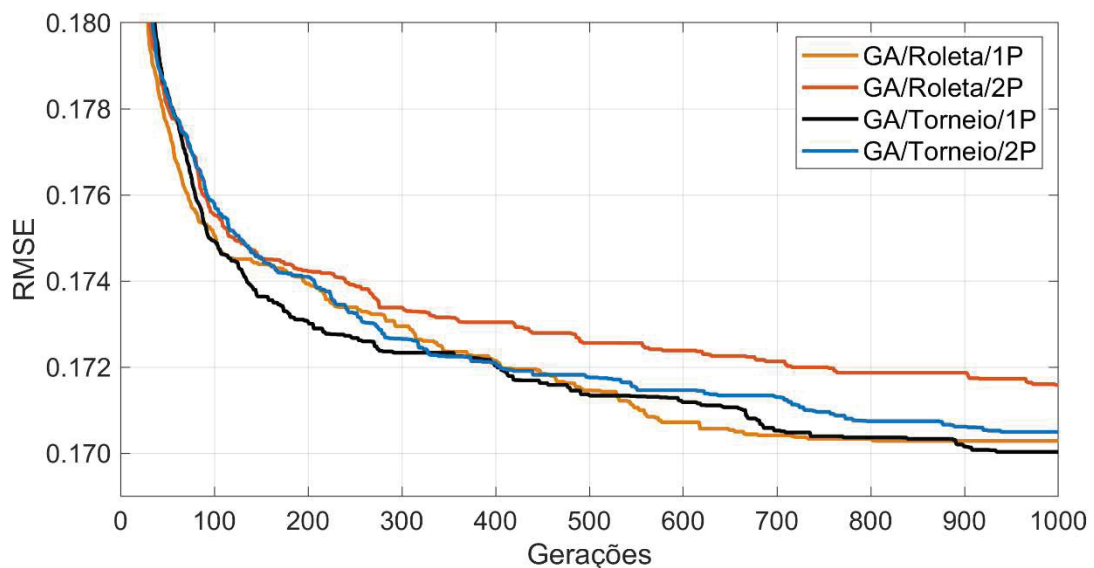


### 6.2.2 Desempenho das técnicas

Utilizando as configurações apresentadas na seção 6.2.1, o algoritmo apresentado no capítulo 5 foi executado para cada uma das 24 opções de técnicas listadas na TABELA 4, sendo realizadas um total de  $S_t = 100$  simulações para cada técnica.

A primeira análise foi realizada para as quatro abordagens do GA, cujas curvas de convergência média das populações de soluções podem ser observadas na FIGURA 16.

FIGURA 16 – Convergência média para o GA (Estudo de caso 1)



FONTE: O autor (2020).

Analisando a FIGURA 16, é possível observar que, em média, a abordagem GA/Torneio/1P atingiu menores valores de  $RMSE$  (0,1700). As abordagens GA/Roletta/1P e GA/Torneio/2P alcançaram desempenhos médios bem próximos a este melhor resultado, apresentando aumentos de 0,15% e 0,27%, respectivamente, enquanto a abordagem GA/Roletta/2P foi a que teve pior desempenho, com um aumento de 0,91%. As quatro abordagens apresentaram velocidades de convergências semelhantes, ou seja, os melhores valores médios obtidos só foram alcançados próximos às últimas gerações. Os resultados estatísticos do  $RMSE$  para

as quatro abordagens do GA estão apresentados na TABELA 6, sendo que o valor médio indicado corresponde ao apresentado na FIGURA 16.

TABELA 6 – Valores de RMSE alcançados pelo GA (Estudo de caso 1)

<b>Abordagem</b>	<b>Mínimo</b>	<b>Médio</b>	<b>Máximo</b>	<b>Desvio Padrão</b>
GA/Roletta/1P	0,1632	0,1703	0,1918	0,008980
GA/Roletta/2P	0,1632	0,1716	0,1918	0,009549
GA/Torneio/1P	<b>0,1632</b>	<b>0,1700</b>	<b>0,1936</b>	<b>0,009013</b>
GA/Torneio/2P	0,1632	0,1705	0,1918	0,008448

FONTE: O autor (2020).

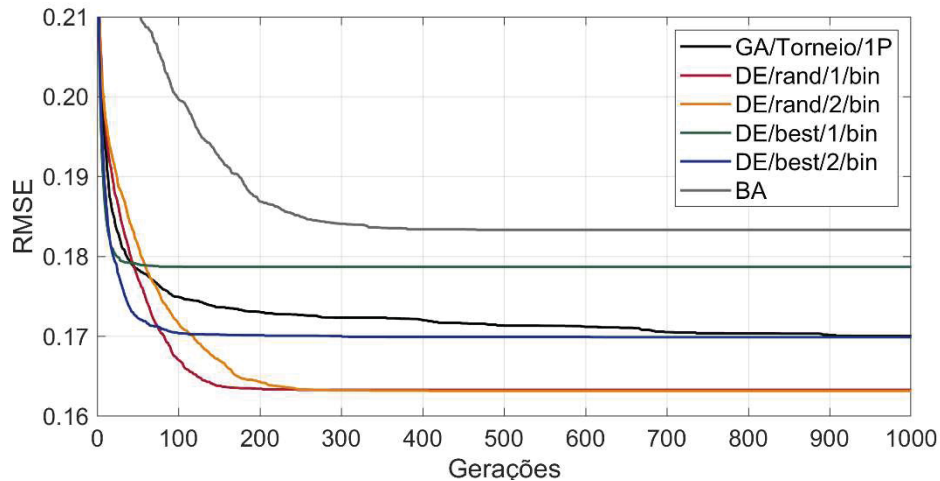
Observando a TABELA 6, é possível perceber que, embora o desempenho médio da abordagem GA/Torneio/1P tenha sido melhor, todas as abordagens foram capazes de alcançar o mesmo valor mínimo para o *RMSE*. Em outras palavras, as quatro abordagens localizaram o modelo que apresenta o *RMSE* da predição de um passo à frente igual a 0,1632. Este foi o melhor modelo encontrado neste estudo de caso e está representado na equação (6.1). Para fins de nomenclatura, este modelo será chamado de CB01.

$$\begin{aligned}
 y(k) = & 120,48 - 1,58y(k-1) - 5,78y(k-3) \\
 & - 31,54u(k-1) - 16,77u(k-3) \\
 & + 0,09y(k-1)y(k-3) + 0,50y(k-1)u(k-1) \quad (6.1) \\
 & + 0,03y^2(k-3) + 0,98y(k-3)u(k-1) \\
 & + 0,51y(k-3)u(k-3) + 3,82u(k-1)u(k-3)
 \end{aligned}$$

Para análise das demais técnicas aplicadas neste trabalho, será utilizada como referência a curva do GA que alcançou o melhor desempenho médio, ou seja, a curva da abordagem GA/Torneio/1P.

Na FIGURA 17 é possível observar o desempenho do BA e das quatro estratégias da DE, quando estes foram associados à codificação binária TF.

FIGURA 17 – Convergência média para BA e DE codificados com TF (Estudo de caso 1)



FONTE: O autor (2020).

Observa-se, pela FIGURA 17, que a DE associada com a codificação binária TF alcançou valores médios menores que o GA com as estratégias DE/rand/1/bin e DE/rand/2/bin, além de apresentar uma convergência média bem mais acelerada. Enquanto a abordagem GA/Torneio/1P alcança o  $RMSE$  médio de 0,1700 somente na geração  $g_G = 934$ , a estratégia DE/rand/1/bin atinge o  $RMSE$  médio de 0,1633 na geração  $g_D = 934$ , a estratégia DE/rand/2/bin chega ao  $RMSE$  médio de 0,1632 na geração  $g_D = 408$ . Vale ressaltar que, com a estratégia DE/rand/2/bin, foi possível obter o modelo CB01 em todas as 100 simulações, conforme valores apresentados na TABELA 7. A estratégia DE/best/2/bin, apesar de apresentar uma convergência média um pouco mais acentuada, alcançou um valor médio de  $RMSE$  muito próximo ao da abordagem GA/Torneio/1P. A estratégia DE/best/1/bin, por sua vez, apresentou um baixo desempenho comparado aos das demais estratégias. No entanto, com todas as estratégias da DE foi possível detectar o modelo CB01 aplicando a codificação binária TF. O BA, por sua vez, ao ser associado à codificação TF apresentou uma convergência média mais lenta em relação às demais técnicas, além de ter atingido um  $RMSE$  médio de 0,1833, como pode ser visto na TABELA 7.

TABELA 7 – Valores de RMSE alcançados por BA e DE com codificação TF

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	0,1632	0,1633	0,1758	0,001262
DE/rand/2/bin	<b>0,1632</b>	<b>0,1632</b>	<b>0,1632</b>	<b>0,000000</b>
DE/best/1/bin	0,1632	0,1787	0,1971	0,007510
DE/best/2/bin	0,1632	0,1699	0,1911	0,007809
BA	0,1658	0,1833	0,2075	0,009401

FONTE: O autor (2020).

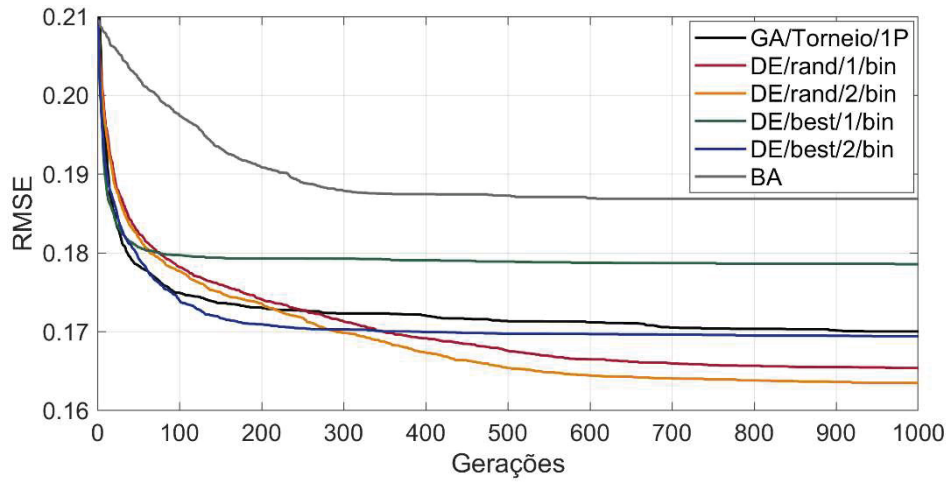
Pela TABELA 7 verifica-se também que o BA associado com a codificação TF não foi capaz de detectar o modelo CB01. O melhor modelo detectado por esta técnica, denominado de CB02, está apresentado na equação (6.2).

$$\begin{aligned}
 y(k) = & 114,49 - 2,00y(k-1) - 4,86y(k-3) \\
 & - 31,61u(k-1) - 14,55u(k-3) \\
 & + 0,11y(k-1)y(k-3) + 0,59y(k-1)u(k-1) \quad (6.2) \\
 & + 0,92y(k-3)u(k-1) + 0,39y(k-3)u(k-3) \\
 & + 3,61u(k-1)u(k-3)
 \end{aligned}$$

Quando utilizada a codificação binária GVP, os desempenhos alcançados pelos algoritmos BA e DE são os que estão apresentados na FIGURA 18. Assim como ocorreu com a codificação TF, as estratégias DE/rand/1/bin e DE/rand/2/bin atingiram valores médios de *RMSE* bem menores que a abordagem GA/Torneio/1P, porém, com uma convergência média mais lenta. Com a codificação GVP, o melhor desempenho médio também foi alcançado pela estratégia DE/rand/2/bin.

A estratégia DE/best/2/bin mais uma vez apresentou desempenho próximo ao da abordagem GA/Torneio/1P. Já a estratégia DE/best/1/bin e o algoritmo BA não apresentaram desempenho satisfatórios comparados com a abordagem GA/Torneio/1P. Os valores estatísticos do *RMSE* obtidos com a codificação GVP estão listados na TABELA 8, onde é possível observar que o modelo CB01 foi detectado por todas as estratégias da DE, mas não pelo BA.

FIGURA 18 – Convergência média para BA e DE codificados com GVP (Estudo de caso 1)



FONTE: O autor (2020).

TABELA 8 – Valores de RMSE alcançados por BA e DE com codificação GVP (Estudo de caso 1)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	0,1632	0,1654	0,1778	0,004387
DE/rand/2/bin	<b>0,1632</b>	<b>0,1635</b>	<b>0,1758</b>	<b>0,001443</b>
DE/best/1/bin	0,1632	0,1786	0,1988	0,008172
DE/best/2/bin	0,1632	0,1694	0,1928	0,007068
BA	0,1740	0,1869	0,2055	0,007400

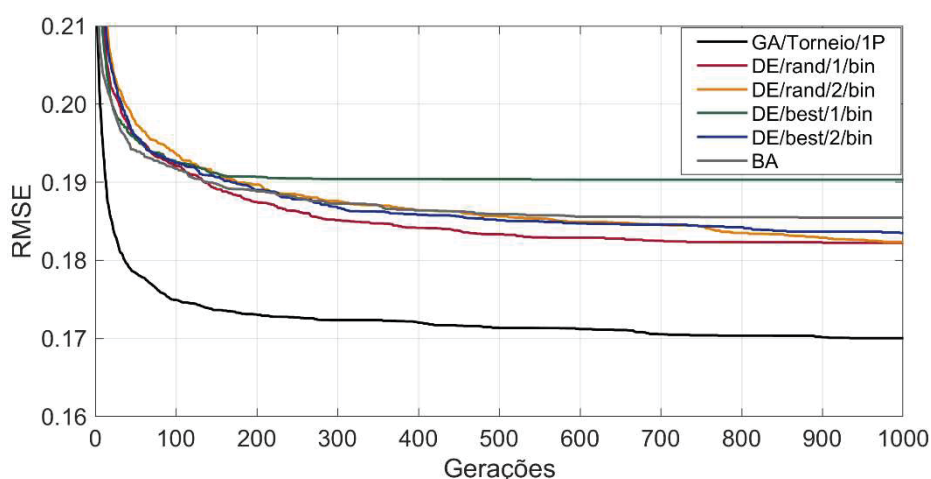
FONTE: O autor (2020).

Pela TABELA 8 verifica-se o melhor modelo detectado pelo BA associado com a codificação GVP possui um  $RMSE = 0,1740$ . Este modelo será denominado de CB03 e está apresentado na equação (6.3).

$$\begin{aligned}
 y(k) = & 141,52 - 1,84y(k-1) - 4,32y(k-3) \\
 & - 36,88u(k-1) - 36,36u(k-3) \\
 & + 0,09y(k-1)y(k-3) + 0,56y(k-1)u(k-1) \\
 & - 0,004y^2(k-2) - 0,66y(k-2)u(k-1) \\
 & + 0,85y(k-2)u(k-3) + 0,006y^2(k-3) \\
 & + 1,74y(k-3)u(k-1) - 0,66y(k-3)u(k-3) \\
 & + 5,13u(k-1)u(k-3) + 4,48u^2(k-3)
 \end{aligned} \tag{6.3}$$

Com a codificação binária AM, a DE e o BA não foram capazes de alcançar resultados médios melhores que a abordagem GA/Torneio/1P, como pode ser visto na FIGURA 19. Os valores estatísticos do *RMSE* obtidos com estas técnicas estão listados na TABELA 9.

FIGURA 19 – Convergência média para BA e DE codificados com AM (Estudo de caso 1)



FONTE: O autor (2020).

TABELA 9 – Valores de RMSE alcançados por BA e DE com codificação AM (Estudo de caso 1)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	<b>0,1632</b>	<b>0,1823</b>	<b>0,1947</b>	<b>0,005990</b>
DE/rand/2/bin	0,1660	0,1824	0,1987	0,005308
DE/best/1/bin	0,1704	0,1903	0,2214	0,008286
DE/best/2/bin	0,1689	0,1835	0,2057	0,006235
BA	0,1678	0,1855	0,1991	0,005511

FONTE: O autor (2020).

Observa-se pela TABELA 9 que, além de ter apresentado o melhor desempenho médio dentro do conjunto analisado, a estratégia DE/rand/1/bin foi a única que conseguiu detectar o modelo CB01. Os melhores modelos encontrados pelas estratégias DE/rand/2/bin, DE/best/1/bin, DE/best/2/bin e pelo BA, com a codificação AM, estão apresentados nas equações (6.4), (6.5), (6.6) e (6.7), respectivamente. Estes modelos receberão as denominações CB04, CB05, CB06 e CB07, respectivamente.

$$\begin{aligned}
y(k) = & 103,91 - 5,94y(k-3) - 29,84u(k-1) \\
& - 12,10u(k-3) + 0,07y(k-1)y(k-3) \\
& - 0,09y(k-1)u(k-3) + 0,03y^2(k-3) \\
& + 1,35y(k-3)u(k-1) + 0,25y(k-3)u(k-3) \\
& + 3,80u(k-1)u(k-3)
\end{aligned} \tag{6.4}$$

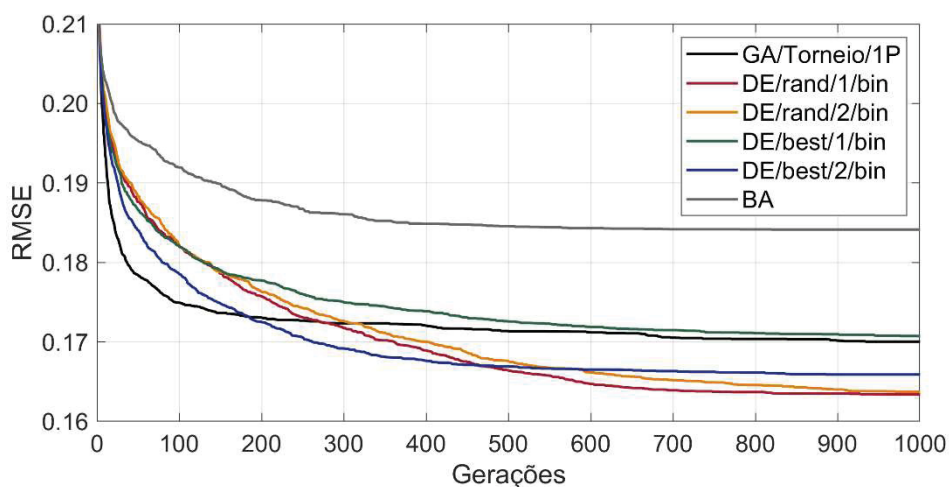
$$\begin{aligned}
y(k) = & 83,71 - 0,07y(k-1) - 5,14y(k-3) - 29,08u(k-1) \\
& + 0,06y(k-1)y(k-3) + 0,03y^2(k-3) \\
& + 1,27y(k-3)u(k-1) + 0,10y(k-3)u(k-3) \\
& + 3,95u(k-1)u(k-3) - 2,49u^2(k-3)
\end{aligned} \tag{6.5}$$

$$\begin{aligned}
y(k) = & 110,57 - 5,79y(k-3) - 36,06u(k-1) \\
& - 12,47u(k-3) + 0,07y(k-1)y(k-3) \\
& - 0,09y(k-1)u(k-3) + 0,03y^2(k-3) \\
& + 1,56y(k-3)u(k-1) + 5,29u(k-1)u(k-3)
\end{aligned} \tag{6.6}$$

$$\begin{aligned}
y(k) = & 104,94 - 5,36y(k-3) - 33,52u(k-1) \\
& - 12,86u(k-3) + 0,06y(k-1)y(k-3) \\
& + 0,03y^2(k-3) + 1,42y(k-3)u(k-1) \\
& + 5,00u(k-1)u(k-3)
\end{aligned} \tag{6.7}$$

Os desempenhos do BA e das estratégias da DE, ao se aplicar a codificação binária MAM, estão apresentados na FIGURA 20. Observa-se uma melhoria considerável nas convergências médias obtidas por todas as estratégias da DE em relação àquelas alcançadas com a codificação AM. As quatro estratégias apresentaram uma redução de 10% no valor do *RMSE* médio. Uma melhoria sutil também pode ser verificada no desempenho do BA, onde a redução apresentada foi de 0,71%.

FIGURA 20 – Convergência média para BA e DE codificados com MAM (Estudo de caso 1)



FONTE: O autor (2020).

Observando a FIGURA 20, verifica-se que a estratégia DE/rand/1/bin continuou obtendo o melhor desempenho médio, porém todas as estratégias da DE conseguiram detectar o modelo CB01. O BA, por sua vez, detectou como melhor modelo o CB05, já apresentado na equação (6.5).

Assim como nas codificações TF e GVP, com a codificação MAM, foi possível obter com as estratégias DE/rand/1/bin, DE/rand/2/bin e DE/best/2/bin, um desempenho melhor que o da abordagem GA/Torneio/1P. Entretanto, vale ressaltar que, embora o *RMSE* médio alcançado pela estratégia DE/best1/bin seja maior que o da abordagem GA/Torneio/1P, houve uma redução expressiva nesta diferença, que caiu de 5%, quando foram aplicadas as codificações TF e GVP, para 0,4%, quando foi aplicada a codificação MAM. Para o caso do BA, aplicando a codificação MAM, esta diferença caiu de 9% e 10%, respectivamente, quando foram aplicadas as codificações AM e GVP, para 8%, mesmo valor de diferença obtido com a codificação TF. Os valores estatísticos do *RMSE* para este caso encontram-se na TABELA 10.

TABELA 10 – Valores de RMSE alcançados por BA e DE com codificação MAM (Estudo de caso 1)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	<b>0,1632</b>	<b>0,1634</b>	<b>0,1678</b>	<b>0,000912</b>
DE/rand/2/bin	0,1632	0,1637	0,1709	0,001456
DE/best/1/bin	0,1632	0,1707	0,1880	0,007052
DE/best/2/bin	0,1632	0,1659	0,1807	0,005177
BA	0,1704	0,1841	0,1958	0,005488

FONTE: O autor (2020).



Além de analisar o desempenho de cada técnica através das curvas de convergência, foi utilizada também a taxa de sucesso definida na equação (5.4) para medir a sua capacidade de localizar um determinado modelo. Como o modelo CB01 foi o que apresentou o menor *RMSE* dentre todos os modelos analisados, ele foi utilizado como referência no cálculo da taxa de localização, *L*, para cada técnica aplicada neste trabalho. Os resultados podem ser observados na TABELA 11.

TABELA 11 – Taxa de localização das técnicas aplicadas na seleção de estrutura (Estudo de caso 1)

Codificação Binária	Estratégia	<i>L</i>	<i>L</i> <sub>MÉDIO</sub> ± desv. pad.
-	GA/Roletas/1P	54%	51,75% ± 0,05
-	GA/Roletas/2P	48%	
-	GA/Torneio/1P	57%	
-	GA/Torneio/2P	48%	
TF	DE/rand/1/bin	99%	50,80% ± 0,48
	DE/rand/2/bin	100%	
	DE/best/1/bin	6%	
	DE/best/2/bin	49%	
	BA	0%	
GVP	DE/rand/1/bin	74%	42,40% ± 0,40
	DE/rand/2/bin	92%	
	DE/best/1/bin	8%	
	DE/best/2/bin	38%	
	BA	0%	
AM	DE/rand/1/bin	2%	0,40% ± 0,01
	DE/rand/2/bin	0%	
	DE/best/1/bin	0%	
	DE/best/2/bin	0%	
	BA	0%	
MAM	DE/rand/1/bin	95%	57,60% ± 0,40
	DE/rand/2/bin	86%	
	DE/best/1/bin	33%	
	DE/best/2/bin	74%	
	BA	0%	

FONTE: O autor (2020).

Observa-se da TABELA 11 que a maior taxa de sucesso foi atingida pela estratégia DE/rand/2/bin associada com a codificação TF, que alcançou o valor máximo possível. O BA, por sua vez, foi a técnica que teve o pior desempenho, apresentando uma taxa de localização de 0% com todas as codificações binárias.

Constata-se também que a estratégia DE/best/1/bin, de modo geral, não apresentou bons resultados quando associada às codificações TF, GVP e AM, mas

seu desempenho melhorou consideravelmente ao ser associada à codificação MAM. Além disso, é possível verificar que a utilização da codificação MAM não apenas melhorou o desempenho de cada uma das estratégias da DE, em relação à codificação AM, como também permitiu atingir uma taxa média de localização maior do que a do GA e de todas as outras codificações binárias.

O custo computacional de cada técnica foi verificado através do tempo de simulação medido em segundos para cada simulação. Os resultados podem ser observados na TABELA 12.

TABELA 12 – Custo computacional de cada técnica utilizada (Estudo de caso 1)

<b>Codificação Binária</b>	<b>Estratégia</b>	<b>Tempo (s) (média ± desv. pad.)</b>
-	GA/Roletta/1P	342 ± 14
-	GA/Roletta/2P	358 ± 13
-	GA/Torneio/1P	352 ± 10
-	GA/Torneio/2P	351 ± 10
TF	DE/rand/1/bin	374 ± 13
	DE/rand/2/bin	373 ± 13
	DE/best/1/bin	373 ± 12
	DE/best/2/bin	374 ± 14
	BA	194 ± 39
GVP	DE/rand/1/bin	358 ± 9
	DE/rand/2/bin	360 ± 12
	DE/best/1/bin	358 ± 10
	DE/best/2/bin	359 ± 10
	BA	390 ± 33
AM	DE/rand/1/bin	371 ± 13
	DE/rand/2/bin	378 ± 16
	DE/best/1/bin	368 ± 13
	DE/best/2/bin	371 ± 15
	BA	492 ± 133
MAM	DE/rand/1/bin	368 ± 10
	DE/rand/2/bin	368 ± 11
	DE/best/1/bin	368 ± 10
	DE/best/2/bin	367 ± 11
	BA	422 ± 31

FONTE: O autor (2020).

Observa-se da TABELA 12 que, com exceção do BA, todas as demais técnicas mantiveram um padrão de tempo médio de simulação situado entre 340s e 380s. O BA obteve tempos médios bastante divergentes para cada tipo de codificação,

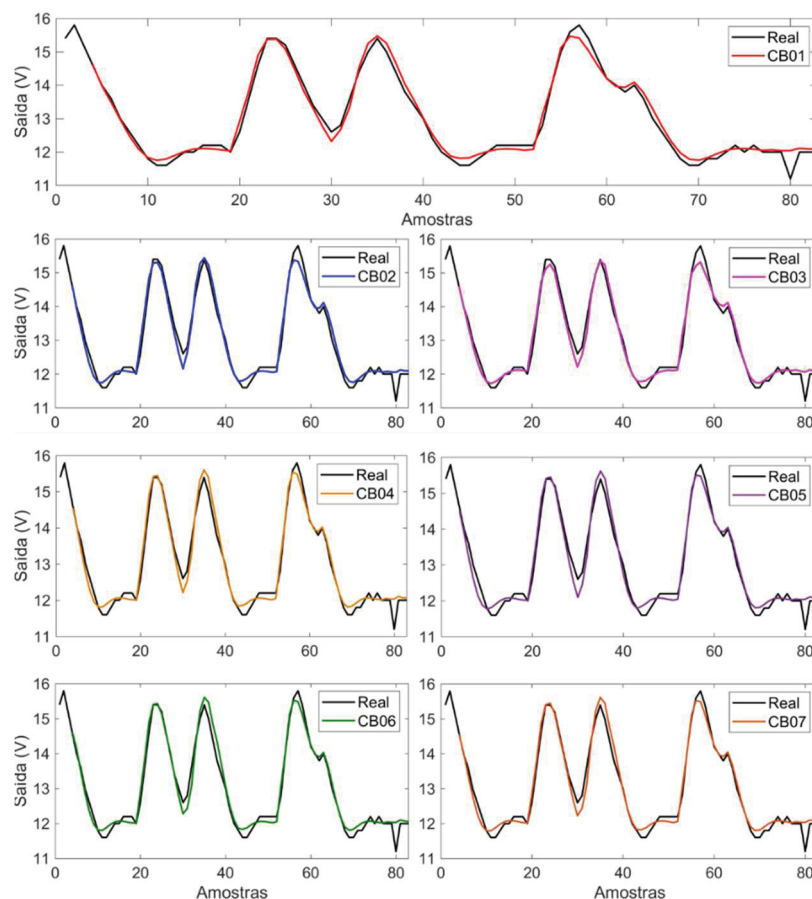
registrando os extremos de menor tempo médio (194s, quando associado com a codificação TF) e maior tempo médio (492s, quando associado com AM).

### 6.2.3 Validação dos modelos obtidos

Nesta seção serão apresentadas as validações dos melhores modelos encontrados para o conversor *buck*, ou seja, os modelos CB01 a CB07. A própria seleção dos modelos já foi feita usando um processo de validação, uma vez que aplica o *RMSE* de previsões de um passo à frente sobre os dados de validação para classificá-los. Entretanto, outras abordagens de validação serão verificadas para averiguar com maior profundidade a qualidade dos modelos apresentados.

As simulações livres dos modelos selecionados encontram-se apresentadas na FIGURA 21. Observa-se que os modelos se ajustam bem aos dados dinâmicos do sistema, apresentando pequenas diferenças entre si em alguns pontos.

FIGURA 21 – Simulação livre dos modelos CB01 a CB07



FONTE: O autor (2020).

Para detectar mais precisamente estas diferenças, foram utilizados os índices  $RMSE_{SL}$  e  $R^2_{SL}$  sobre os dados de validação, cujos valores podem ser visualizados na TABELA 13. Nesta tabela também foram incluídos os mesmos índices calculados sobre os dados de estimação.

TABELA 13 – Índices de desempenho para os modelos CB01 a CB07

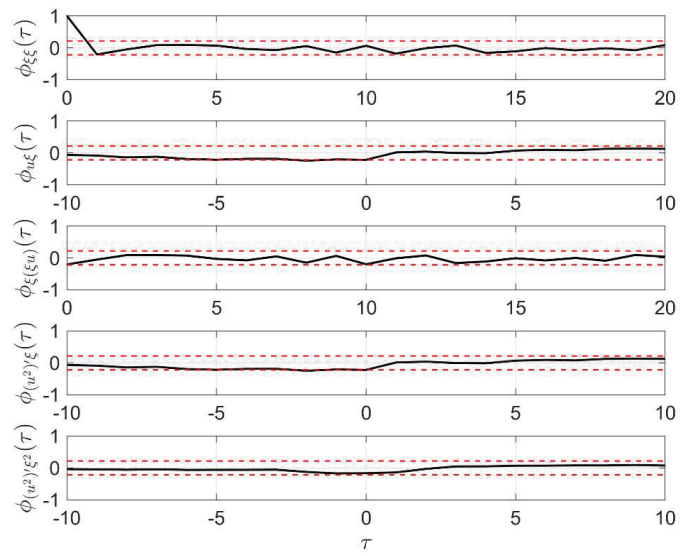
Modelo	Estimação		Validação	
	$RMSE_{SL}$	$R^2_{SL}$	$RMSE_{SL}$	$R^2_{SL}$
CB01	<b>0,1986</b>	<b>97,50%</b>	<b>0,1863</b>	<b>97,79%</b>
CB02	0,2070	97,29%	0,2117	97,15%
CB03	0,2002	97,46%	0,2023	97,39%
CB04	0,1933	97,63%	0,2095	97,20%
CB05	0,1943	97,61%	0,2268	96,72%
CB06	0,1922	97,66%	0,2158	97,04%
CB07	0,1943	97,61%	0,2084	97,23%

FONTE: O autor (2020).

Através da TABELA 13, verifica-se que o modelo CB01 apresenta melhores valores para os índices  $RMSE_{SL}$  e  $R^2_{SL}$ , calculados sobre os dados de validação. Portanto, segundo estes critérios, o mesmo pode ser considerado o modelo que melhor se ajusta aos dados dinâmicos do sistema ao utilizar a simulação livre. Vale ressaltar que, caso haja uma tolerância nos valores destes índices, alguns dos demais modelos podem ser utilizados, já que apresentam, sob o ponto de vista da complexidade, a vantagem de terem menos termos.

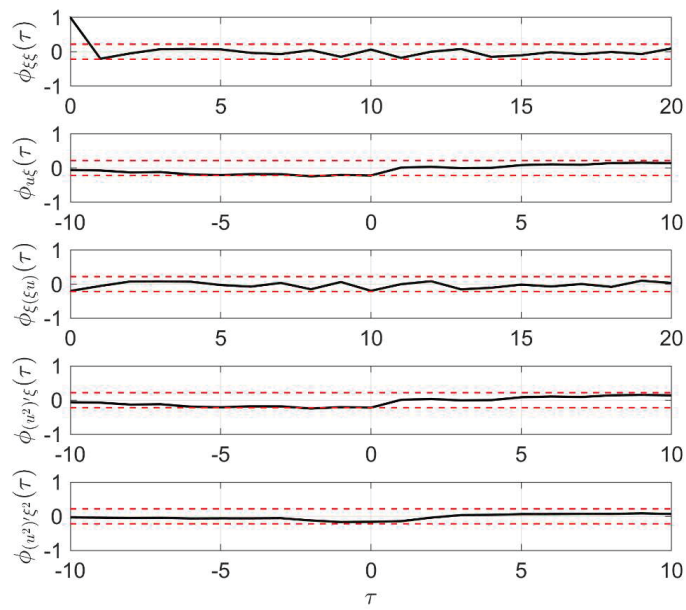
O próximo método aplicado para avaliar os modelos é a validação estatística, onde será possível analisar o comportamento dos resíduos e verificar se estes são aleatórios, assemelhando-se a um ruído branco. Nas FIGURAS 22 a 28 é possível observar as análises de resíduos para os modelos CB01 a CB07, respectivamente.

FIGURA 22 – Validação estatística do modelo CB01.



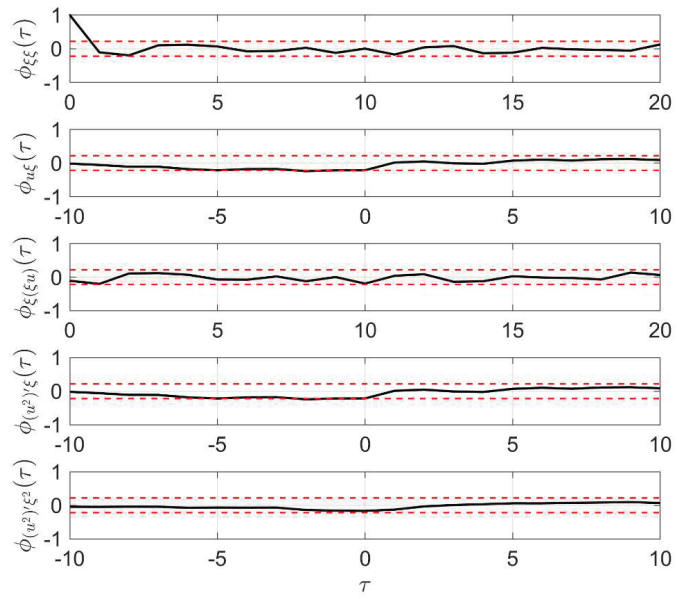
FONTE: O autor (2020).

FIGURA 23 – Validação estatística do modelo CB02.



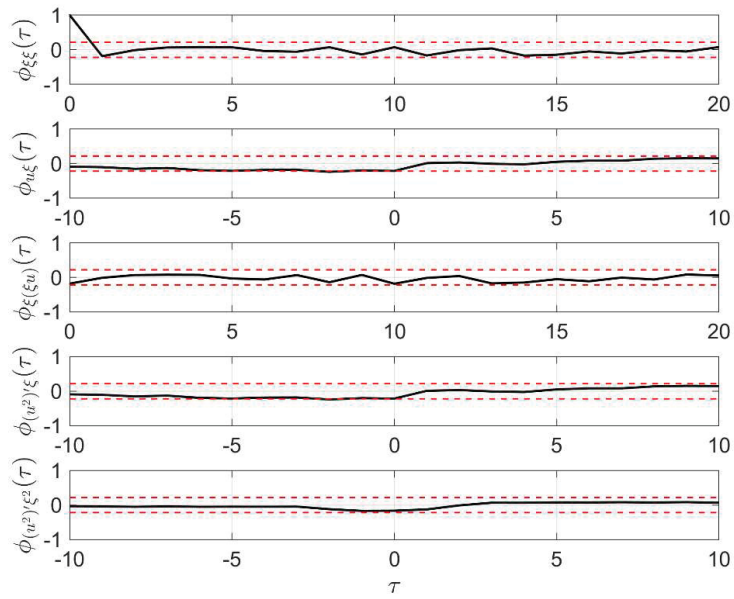
FONTE: O autor (2020).

FIGURA 24 – Validação estatística do modelo CB03.



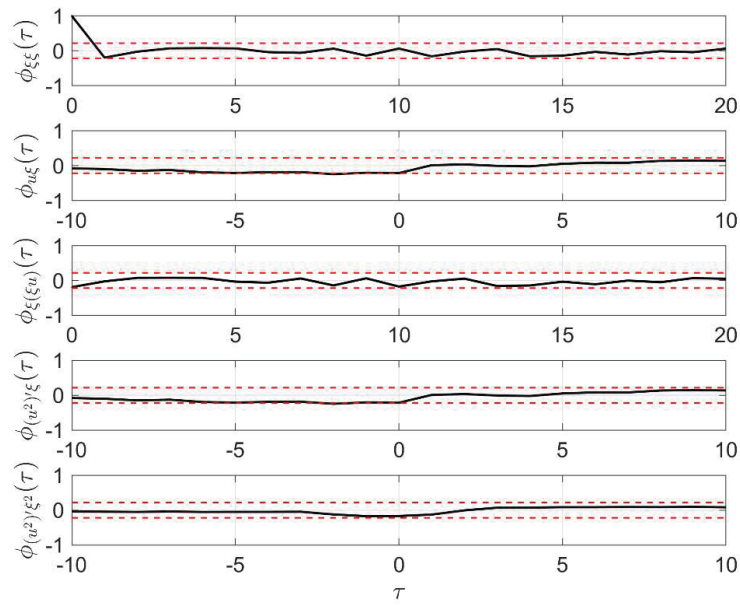
FONTE: O autor (2020).

FIGURA 25 – Validação estatística do modelo CB04.



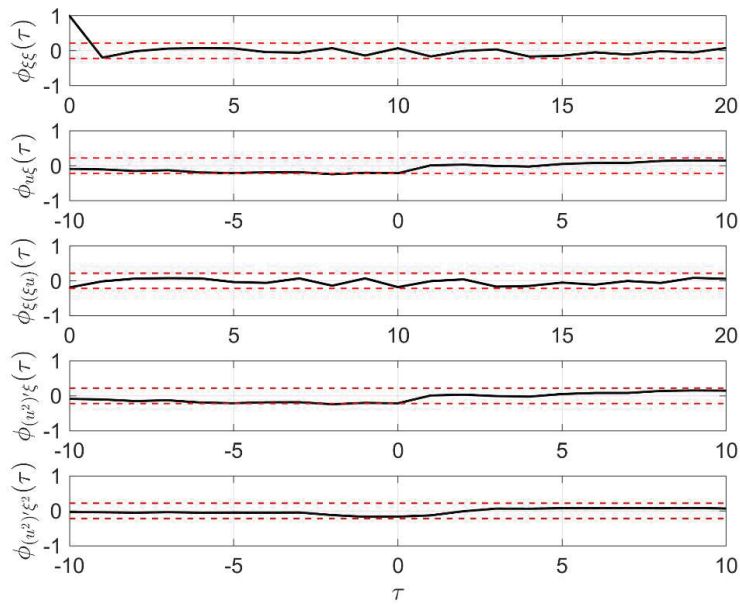
FONTE: O autor (2020).

FIGURA 26 – Validação estatística do modelo CB05.



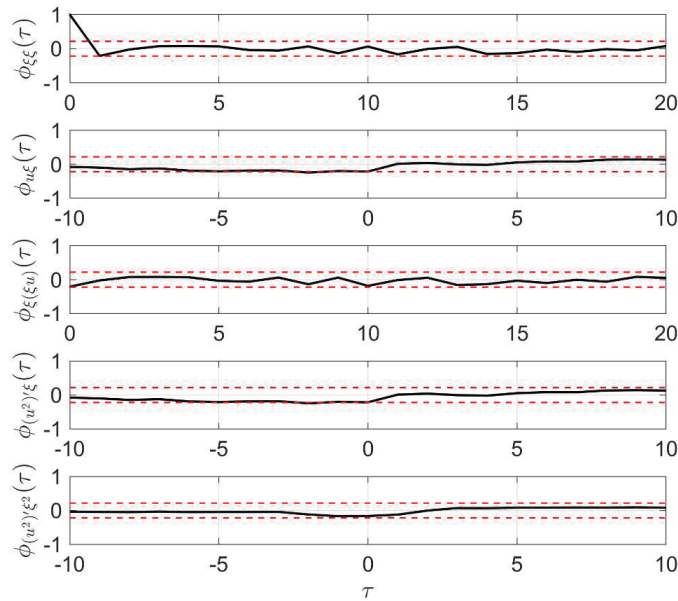
FONTE: O autor (2020).

FIGURA 27 – Validação estatística do modelo CB06.



FONTE: O autor (2020).

FIGURA 28 – Validação estatística do modelo CB07.



FONTE: O autor (2020).

Analisando as FIGURAS 22 a 28, percebe-se que as funções de autocorrelação e correlação cruzada encontram-se dentro do intervalo de confiança de 95% (linhas vermelhas tracejadas), para todos os modelos apresentados. Isto sugere que, em todos os casos, os resíduos possuem comportamento aleatório e aproximam-se de um ruído branco. Além disso, também é possível concluir que os resíduos não são correlacionados com a entrada, não contendo informação útil a respeito dos dados utilizados no procedimento de identificação. Desta forma, é possível concluir que todos os modelos apresentados representam bem o sistema, embora suas simulações livres apresentem desempenhos razoavelmente diferentes, como visto na TABELA 13.

Além das análises já mencionadas foram verificadas a complexidade dos modelos e a presença de determinados regressores em suas estruturas. As informações referentes à quantidade de termos em cada modelo e ao número de ocorrências de cada termos nos modelos CB01 a CB07 foram consolidadas e estão apresentadas na FIGURA 29. Observa-se que o modelo com menos termos é o modelo CB07 (8 termos) e o modelo com mais termos é o modelo CB03 (15 termos). Alguns regressores podem ser considerados com elevado grau de importância, uma vez que ocorreram em todos os modelos. São eles:  $y(k-3)$ ,  $u(k-1)$ ,  $y(k-1)y(k-3)$ ,  $y(k-3)u(k-1)$ ,  $u(k-1)u(k-3)$ , assim como a



constante. os quais ocorreram em todos os modelos. Outros regressores que podem ser considerados relevantes são os que ocorreram na maioria dos modelos, que são os termos  $u(k-3)$ ,  $y^2(k-3)$  (ambos com 6 ocorrências) e  $y(k-3)u(k-3)$  (5 ocorrências). Também é possível verificar pela FIGURA 29 que diversos regressores não apresentaram relevância, segundo as técnicas utilizadas, uma vez que não apareceram em nenhum dos modelos identificados (0 ocorrências).

FIGURA 29 – Análise dos regressores dos modelos CB01 a CB07.

		CB01	CB02	CB03	CB04	CB05	CB06	CB07	OCORRÊNCIAS DOS TERMOS
QUANTIDADE DE TERMOS		11	10	15	10	10	9	8	
1º	1	x	x	x	x	x	x	x	7
2º	$y(k-1)$	x	x	x		x			4
3º	$y(k-2)$								0
4º	$y(k-3)$	x	x	x	x	x	x	x	7
5º	$u(k-1)$	x	x	x	x	x	x	x	7
6º	$u(k-2)$								0
7º	$u(k-3)$	x	x	x	x		x	x	6
8º	$y^2(k-1)$								0
9º	$y(k-1)y(k-2)$								0
10º	$y(k-1)y(k-3)$	x	x	x	x	x	x	x	7
11º	$y(k-1)u(k-1)$	x	x	x					3
12º	$y(k-1)u(k-2)$								0
13º	$y(k-1)u(k-3)$				x		x		2
14º	$y(k-2)y(k-3)$			x					1
15º	$y^2(k-2)$								0
16º	$y(k-2)u(k-1)$			x					1
17º	$y(k-2)u(k-2)$								0
18º	$y(k-2)u(k-3)$			x					1
19º	$y^2(k-3)$	x		x	x	x	x	x	6
20º	$y(k-3)u(k-1)$	x	x	x	x	x	x	x	7
21º	$y(k-3)u(k-2)$								0
22º	$y(k-3)u(k-3)$	x	x	x	x	x			5
23º	$u^2(k-1)$								0
24º	$u(k-1)u(k-2)$								0
25º	$u(k-1)u(k-3)$	x	x	x	x	x	x	x	7
26º	$u^2(k-2)$								0
27º	$u(k-2)u(k-3)$								0
28º	$u^2(k-3)$			x		x			2

FONTE: O autor (2020).

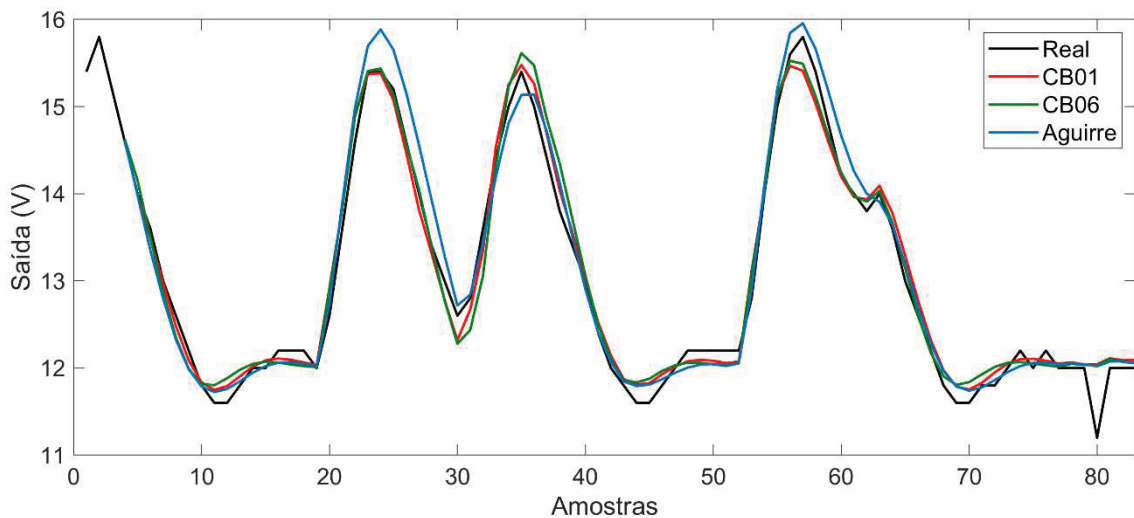
#### 6.2.4 Comparações com a literatura

Os dados do conversor *buck* apresentado neste estudo de caso foram utilizados em outros trabalhos que também aplicaram técnicas de modelagem caixa preta. Aguirre, Donoso-Garcia e Santos-Filho (2000) aplicaram MQ ortogonais associado ao ERR e obtiveram alguns modelos não lineares. O que apresentou maior adequação aos dados dinâmicos do sistema, está apresentado na equação (6.8).

$$\begin{aligned}
 y(k) = & 1,1101y(k-1) - 3,5707y(k-2) \\
 & + 5,5123 \times 10^{-2}y(k-1)^2 - 2,6830u^2(k-1) \\
 & - 9,8706 \times 10^{-2}y(k-1)y(k-3) \\
 & + 9,2757 \times 10^{-2}y(k-2)y(k-3) \\
 & - 8,1247 \times 10^{-4}y(k-3)u(k-1) + 27,632 \\
 & + 0,78813y(k-2)u(k-1)
 \end{aligned} \tag{6.8}$$

Modelos obtidos neste estudo de caso foram comparados com o modelo de Aguirre através de simulação livre e cálculo dos índices  $RMSE_{SL}$  e  $R_{SL}^2$ . Os resultados desta comparação podem ser observados na FIGURA 30 e na TABELA 14. Para esta comparação, somente os dados de validação foram utilizados.

FIGURA 30 – Simulação livre dos modelos CB01, CB06 e Aguirre



FONTE: O autor (2020).

TABELA 14 – Índices de desempenho para os modelos CB01, CB06 e Aguirre

<b>Modelo</b>	<b><math>RMSE_{SL}</math></b>	<b><math>R_{SL}^2</math></b>
CB01	0,1863	97,79%
CB06	0,2158	97,04%
Aguirre	0,2329	96,55%

FONTE: O autor (2020).

A comparação foi realizada apenas com dois modelos pelos seguintes motivos: o modelo CB01 é o que apresenta o melhor desempenho no quesito ajuste aos dados dinâmicos, no entanto apresenta 11 termos, ou seja mais termos que o modelo de Aguirre; o modelo CB06, por sua vez, é o que apresenta a mesma quantidade de termos do modelo de Aguirre, possuindo 9 termos.

O modelo CB01 possui um melhor desempenho apresentando o  $RMSE_{SL}$  20% menor e o  $R_{SL}^2$  1,24% maior, em relação ao modelo de Aguirre. Entretanto, é possível reduzir a credibilidade do desempenho do modelo CB01 quando observada a sua complexidade, uma vez que o modelo de Aguirre apresenta menos termos.

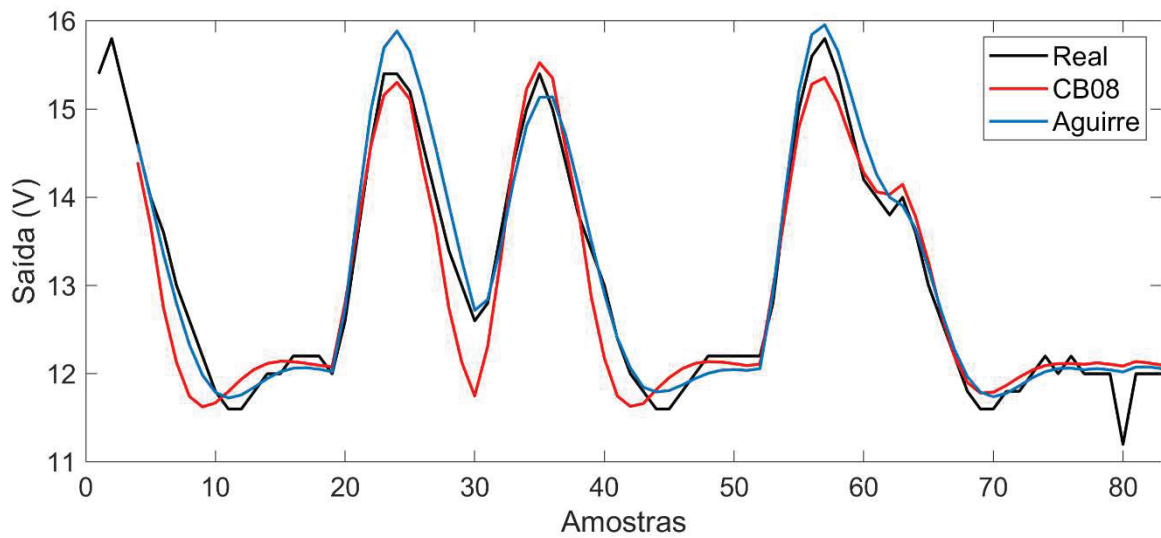
Neste caso, observa-se então o desempenho do modelo CB06, o qual apresenta o  $RMSE_{SL}$  7,34% menor e o  $R_{SL}^2$  0,5% maior que o modelo de Aguirre, mesmo possuindo a mesma quantidade de termos.

Analisando melhor o caso da complexidade do modelo CB01, observa-se que é possível reduzi-lo para 9 termos, afim de efetuar uma comparação mais justa com o modelo de Aguirre. Basta fazer uso de alguma técnica eliminativa de seleção de estrutura e determinar quais termos deveriam ser excluídos do modelo. Um procedimento simples consiste em eliminar os termos do modelo que tenham coeficientes bem menores do que os demais (KADTKE; BRUSH; HOLZFUSS, 1993). No entanto, Aguirre (1994) alerta para o fato de que este método é por demais simplista e, para modelos não lineares, não pode ser considerado como eficiente em geral.

De fato, observando o modelo CB01 através da equação (6.1), verifica-se que os termos candidatos a serem excluídos, com base nesta regra, seriam  $y(k-1)y(k-3)$  e  $y^2(k-3)$ . Efetuando este procedimento e estimando novamente os parâmetros, obtém-se o modelo da equação (6.9), denominado como CB08, o qual apresenta uma considerável redução de desempenho, conforme observado na FIGURA 31 e na TABELA 15.

$$\begin{aligned}
y(k) = & 53,68 + 0,68y(k-1) - 1,35y(k-3) - 27,67u(k-1) \\
& - 2,67u(k-3) + 0,07y(k-1)u(k-1) \\
& + 1,05y(k-3)u(k-1) - 0,60y(k-3)u(k-3) \\
& + 4,26u(k-1)u(k-3)
\end{aligned} \tag{6.9}$$

FIGURA 31 – Simulação livre dos modelos CB08 e Aguirre



FONTE: O autor (2020).

TABELA 15 – Índices de desempenho para o modelo CB08

Modelo	$RMSE_{SL}$	$R^2_{SL}$
CB08	0,3412	92,59%

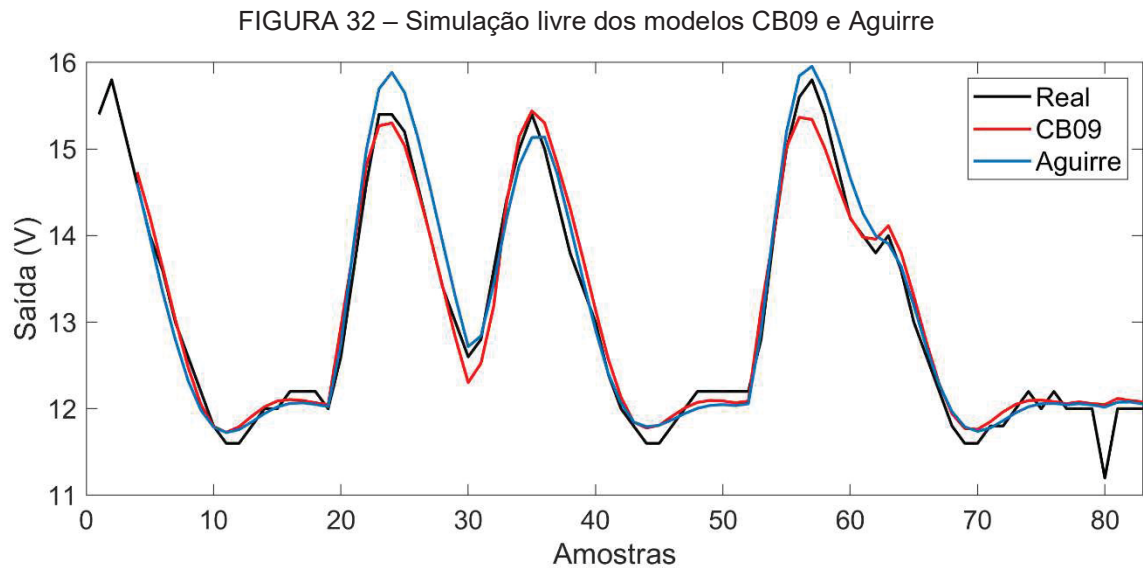
FONTE: O autor (2020).

Para contornar esta problemática, optou-se pela aplicação de uma regra direcionada ao ajuste do modelo aos dados dinâmicos, em vez de apenas observar os valores dos coeficientes de cada termo. Seguindo uma filosofia semelhante ao ERR, buscou-se calcular o impacto de cada termo com base em algum índice. Enquanto o ERR permite quantificar a redução do erro devido à inclusão de cada termo, configurando uma estratégia construtiva de seleção de estrutura, a regra aqui adotada toma o caminho oposto. Calcula-se o aumento do erro devido à exclusão de cada termo de um modelo experimental, neste caso o modelo CB01. A medida de erro adotada foi o  $RMSE$ .

Seguindo esta regra, constatou-se que os termos que deveriam ser excluídos do modelo CB01 são  $y^2(k-3)$  e  $y(k-3)u(k-3)$ . Assim sendo, excluíram-se os termos indicados e efetuou-se novamente a estimação dos parâmetros, gerando o modelo CB09, dado por:

$$\begin{aligned}
 y(k) = & 121,98 - 1,78y(k-1) - 4,65y(k-3) \\
 & - 40,44u(k-1) - 14,57u(k-3) \\
 & + 0,10y(k-1)y(k-3) + 0,53y(k-1)u(k-1) \\
 & + 1,27y(k-3)u(k-1) + 5,80u(k-1)u(k-3)
 \end{aligned} \quad (6.10)$$

Na FIGURA 32 e na TABELA 16, pode ser verificado o desempenho do modelo CB09, o qual possui o mesmo número de termos que o modelo de Aguirre, possibilitando, desta forma, uma comparação mais aceitável do ponto de vista da complexidade do modelo.



FONTE: O autor (2020).

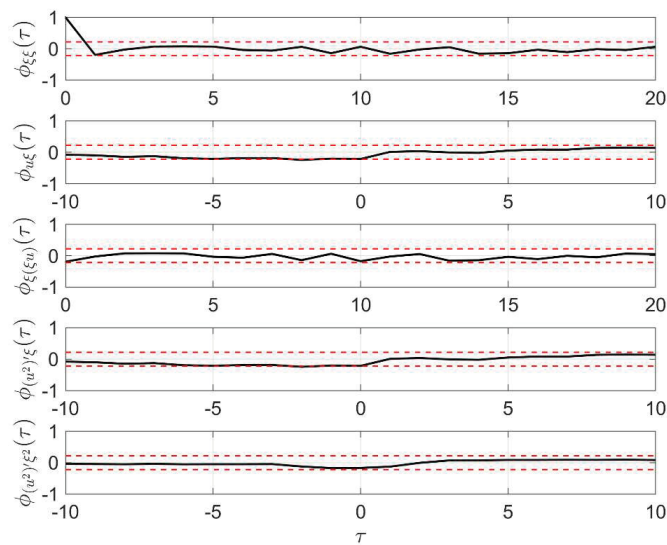
TABELA 16 – Índices de desempenho para o modelo CB09

Modelo	$RMSE_{SL}$	$R^2_{SL}$
CB09	0,2081	97,24%

FONTE: O autor (2020).

Observa-se que o modelo CB09 apresenta o  $RMSE_{SL}$  10,64% menor e o  $R_{SL}^2$  0,7% maior, em relação ao modelo de Aguirre. Vale frisar que, com exceção dos modelos CB01 e CB03, o modelo CB09 apresentou ainda um desempenho melhor que os dos demais modelos obtidos neste estudo de caso, situando-se como o terceiro melhor modelo encontrado. Assim como foi feito para os modelos CB01 a CB07, efetuou-se também a análise de resíduos do modelo CB09, que pode ser observada na FIGURA 33.

FIGURA 33 – Validação estatística do modelo CB09.



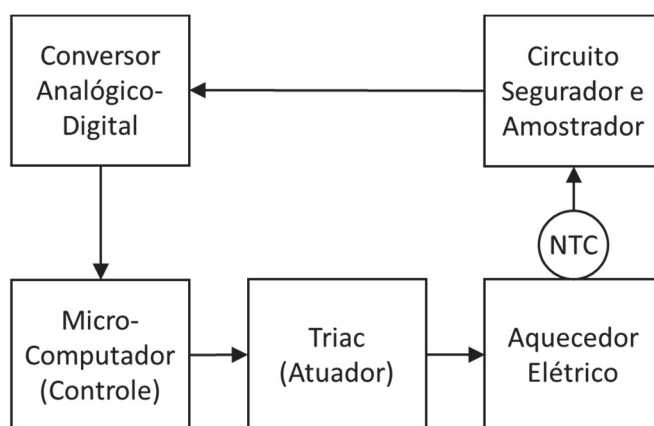
FONTE: O autor (2020).

Verifica-se da FIGURA 33 que, no caso do modelo CB09, as funções de autocorrelação e correlação cruzada também permaneceram dentro do intervalo de confiança de 95%.

### 6.3 ESTUDO DE CASO 2: AQUECEDOR ELÉTRICO

Nesta seção será apresentado o aquecedor elétrico utilizado neste estudo de caso. Um dos primeiros trabalhos que trata da modelagem caixa-preta deste sistema é o estudo de Rodrigues (1996). Além deste, outros trabalhos na área de identificação fazem uso dos dados deste sistema (AGUIRRE, 2015; CORRÊA, 1997). O diagrama do de blocos do aquecedor elétrico é apresentado na FIGURA 34.

FIGURA 34 – Diagrama de blocos do aquecedor elétrico



FONTE: (RODRIGUES, 1996).

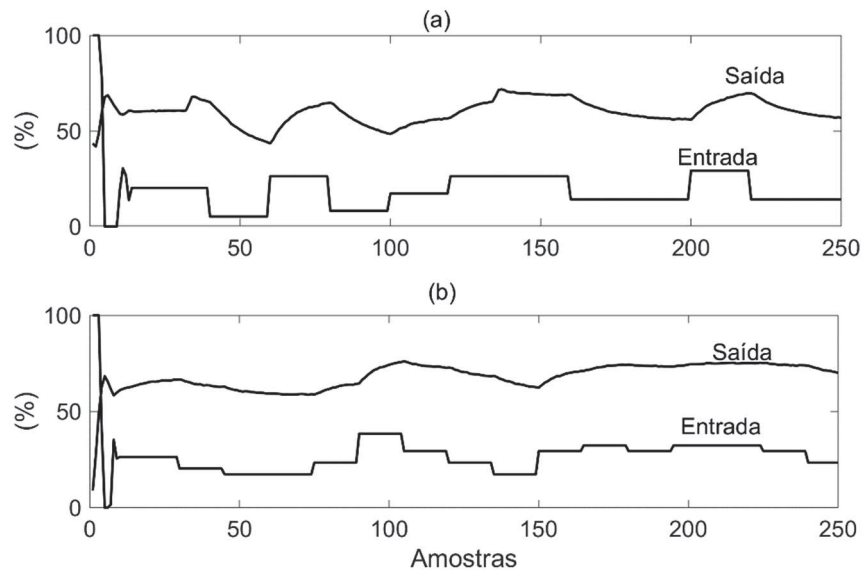
Este sistema consiste em uma caixa metálica com uma lâmpada elétrica de 200W no seu interior atuando como elemento de aquecimento interno. A temperatura da superfície da caixa é medida e informada a um microcomputador, onde é feito o controle digital da temperatura. O sensor utilizado para medição de temperatura é um resistor com coeficiente de temperatura negativo (NTC) ligado em ponte de Wheatstone, a qual encontra-se equilibrada à temperatura ambiente. O sinal de tensão resultante do desbalanceamento da ponte é processado por um conversor analógico-digital de 12 bits e por um microcomputador, onde são implementados os algoritmos de controle. O sinal de controle gerado é aplicado à planta, conforme visto na FIGURA 34. O elemento atuador é um triac operando como chave. O objetivo de controle é ajustar a temperatura de um ponto na superfície externa do aquecedor a um perfil de referência previamente estabelecido. O controlador determina o tempo em que a lâmpada permanece acesa entre duas amostragens, gerando um sinal de controle normalizado entre 0% e 100% (RODRIGUES, 1996).

No procedimento de aquisição de dados do aquecedor, a planta operou em malha fechada com um controlador automático proporcional-integral-derivativo (PID) até que um dado perfil de referência fosse ajustado. Em seguida, a malha de realimentação foi aberta, o sinal de teste aplicado ao comando do triac e o correspondente sinal de temperatura foi medido e processado. O sinal de teste utilizado para excitar a dinâmica do sistema foi do tipo ruído quantizado, gerado através de um algoritmo que sorteia aleatoriamente a amplitude do sinal e a mantém constante durante um intervalo de tempo previamente escolhido (pelo usuário). Ao final deste intervalo, o algoritmo modifica a amplitude do sinal de maneira quantizada



e a mantém constante durante o mesmo intervalo de tempo anterior (RODRIGUES, 1996). Na FIGURA 35, observam-se duas massas de dados obtidas durante a experimentação do aquecedor elétrico. O sinal de entrada mostrado é a tensão aplicada ao comando do triac que controla o estado da lâmpada e a saída mostra a temperatura da superfície externa do aquecedor, normalizada em relação a uma temperatura máxima possível.

FIGURA 35 – Dados coletados do aquecedor elétrico. (a) sinal de entrada e (b) sinal de saída.



FONTE: (RODRIGUES, 1996).

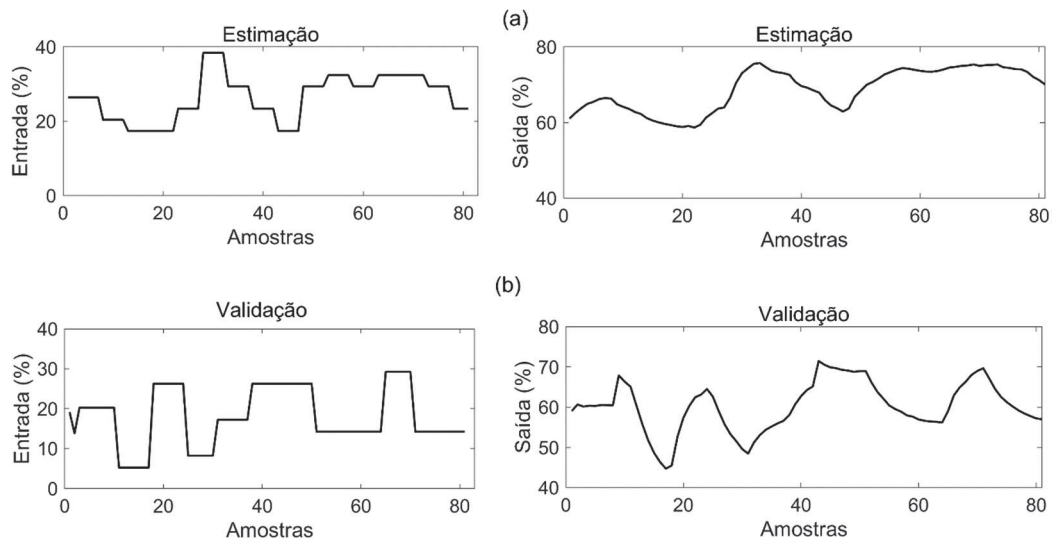
Os dados apresentados na FIGURA 35 são adimensionais e as escalas verticais dos gráficos indicam porcentagens dos valores máximos da temperatura do aquecedor, considerada como sinal de saída,  $y(k)$ , e de tensão aplicada ao triac, adotada como sinal de entrada,  $u(k)$ . O período de amostragem dos sinais é  $T_s = 70s$  (RODRIGUES, 1996).

O sinal de saída da massa de dados da FIGURA 35(a) apresenta variações bruscas nas amostras  $k \approx 30$  e  $k \approx 130$ , as quais não refletem mudanças no sinal de entrada. Acredita-se que estas variações ocorram por falhas de comunicação entre o microcomputador e o processo, constituindo condições anormais de funcionamento. Desta forma, foi utilizada a massa de dados apresentada na FIGURA 35(b) para derivar modelos, enquanto a massa de dados da FIGURA 35(a) foi aplicada na validação dos modelos identificados.



Os dados de identificação do aquecedor elétrico foram obtidos selecionando-se um a cada três pontos das massas de dados apresentadas na FIGURA 35, o que resulta em um período de amostragem final de  $T_s^* = 210s$ . Importante frisar que as primeiras amostras não foram utilizadas, uma vez que o sistema operou em malha fechada até a amostra  $k = 10$ , aproximadamente (RODRIGUES, 1996). A partir destas considerações, foram formados os conjuntos de estimação e validação aplicados neste estudo de caso, os quais podem ser observados na FIGURA 36. Neste caso, as variações bruscas que estavam presentes na FIGURA 35(a) agora ficam situadas nas amostras  $k \approx 10$  e  $k \approx 43$ , do sinal de saída apresentado na FIGURA 36(b).

FIGURA 36 – Sinais de entrada e saída do aquecedor elétrico. Dados de estimação (a) e validação (b).



FONTE: O autor (2020).

Os dados apresentados na FIGURA 36 foram utilizados por Rodrigues (1996) para obtenção de modelos para o aquecedor elétrico. Estes mesmos dados foram disponibilizados na internet pelo autor, o que permitiu sua utilização neste estudo de caso.

### 6.3.1 Parâmetros do algoritmo

Inicialmente foi necessário selecionar os parâmetros relacionados ao conjunto de termos candidatos e às metaheurísticas. Com estes valores escolhidos, foi possível aplicar o algoritmo apresentado no capítulo 5.

O conjunto de termos candidatos foi gerado considerando  $n_y = 2$ ,  $n_u = 2$ ,  $l = 3$  e  $d = 1$ , seguindo o mesmo procedimento adotado por Rodrigues (1996). Foram adotados os mesmos valores apresentados pelo autor para que a comparação entre o modelo por ele apresentado e os modelos obtidos neste trabalho ocorresse de forma mais coerente. Os termos possíveis para compor os modelos estão apresentados na FIGURA 37.

FIGURA 37 – Conjunto de termos candidatos para geração de modelos do aquecedor elétrico.

1º	1	10º	$y^2(k-2)$	19º	$y^2(k-1)u(k-2)$	28º	$y^2(k-2)u(k-2)$
2º	$y(k-1)$	11º	$y(k-2)u(k-1)$	20º	$y(k-1)y^2(k-2)$	29º	$y(k-2)u^2(k-1)$
3º	$y(k-2)$	12º	$y(k-2)u(k-2)$	21º	$y(k-1)y(k-2)u(k-1)$	30º	$y(k-2)u(k-1)u(k-2)$
4º	$u(k-1)$	13º	$u^2(k-1)$	22º	$y(k-1)y(k-2)u(k-2)$	31º	$y(k-2)u^2(k-2)$
5º	$u(k-2)$	14º	$u(k-1)u(k-2)$	23º	$y(k-1)u^2(k-1)$	32º	$u^3(k-1)$
6º	$y^2(k-1)$	15º	$u^2(k-2)$	24º	$y(k-1)u(k-1)u(k-2)$	33º	$u^2(k-1)u(k-2)$
7º	$y(k-1)y(k-2)$	16º	$y^3(k-1)$	25º	$y(k-1)u^2(k-2)$	34º	$u(k-1)u^2(k-2)$
8º	$y(k-1)u(k-1)$	17º	$y^2(k-1)y(k-2)$	26º	$y^3(k-2)$	35º	$u^3(k-2)$
9º	$y(k-1)u(k-2)$	18º	$y^2(k-1)u(k-1)$	27º	$y^2(k-2)u(k-1)$		

FONTE: O autor (2020).

A quantidade de termos candidatos é o que irá determinar o tamanho dos vetores binários manipulados pelas metaheurísticas. Ou seja, neste caso tem-se  $N_{V_G} = N_{V_D} = N_{V_B} = 35$ . Para esta quantidade de termos candidatos, existem  $2^{35} - 1 = 34.359.738.367$  estruturas possíveis de serem utilizadas. Para o tamanho das populações das metaheurísticas foi adotado o valor  $N_{P_G} = N_{P_D} = N_{P_B} = 150$ . O número máximo de gerações adotado foi  $g_{G_{max}} = g_{D_{max}} = g_{B_{max}} = 1000$ , considerado como critério de parada do algoritmo.

Os valores de parâmetros específicos de cada metaheurística podem ser observados na TABELA 17. Estes valores foram determinados heurísticamente

através de testes independentes, seguindo o mesmo procedimento descrito na seção 6.2.1, onde foram apresentados os detalhes de tais testes.

TABELA 17 – Parâmetros das metaheurísticas (Estudo de caso 2)

Metaheurística	Parâmetro	Valor
GA	$t_{sn}$	0,6
	$t_c$	0,5
	$p_m$	0,05
DE	$F_m$	0,7
	$C_r$	0,3
BA	$\eta$	0,7
	$\vartheta$	0,3

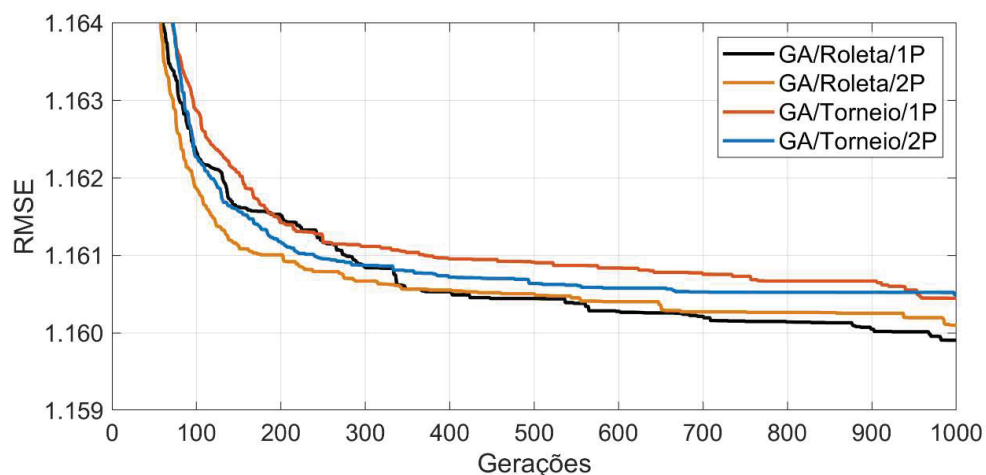
FONTE: O autor (2020).

### 6.3.2 Desempenho das técnicas

Utilizando as configurações apresentadas na seção 6.3.1, o algoritmo apresentado no capítulo 5 foi executado para cada uma das 24 opções de técnicas listadas na TABELA 4, sendo realizadas um total de  $S_t = 100$  simulações para cada técnica.

Assim como no estudo de caso apresentado na seção 6.2, a primeira análise foi realizada para as quatro abordagens do GA, cujas curvas de convergência média das populações de soluções podem ser observadas na FIGURA 38.

FIGURA 38 – Convergência média para o GA (Estudo de caso 2)



FONTE: O autor (2020).

Analisando a FIGURA 38, é possível observar que, em média, a abordagem GA/Roletta/1P atingiu menores valores de *RMSE*. A abordagem GA/Roletta/2P foi a que alcançou um desempenho mais próximo a este melhor resultado, com um aumento de 0,02%, e as outras duas abordagens tiveram desempenho próximos um do outro, porém piores, apresentando ambas um aumento de 0,05%. As quatro abordagens apresentaram velocidades de convergências semelhantes, ou seja, os melhores valores médios obtidos só foram alcançados próximos às últimas gerações. Os resultados estatísticos do *RMSE* para as quatro abordagens do GA estão apresentados na TABELA 18, sendo que o valor médio indicado corresponde ao apresentado na FIGURA 38.

TABELA 18 – Valores de RMSE alcançados pelo GA (Estudo de caso 2)

<b>Abordagem</b>	<b>Mínimo</b>	<b>Médio</b>	<b>Máximo</b>	<b>Desvio Padrão</b>
GA/Roletta/1P	<b>1,1516</b>	<b>1,1599</b>	<b>1,1929</b>	<b>0,005784</b>
GA/Roletta/2P	1,1516	1,1601	1,1721	0,005117
GA/Torneio/1P	1,1516	1,1604	1,1797	0,004859
GA/Torneio/2P	1,1516	1,1605	1,1717	0,004622

FONTE: O autor (2020).

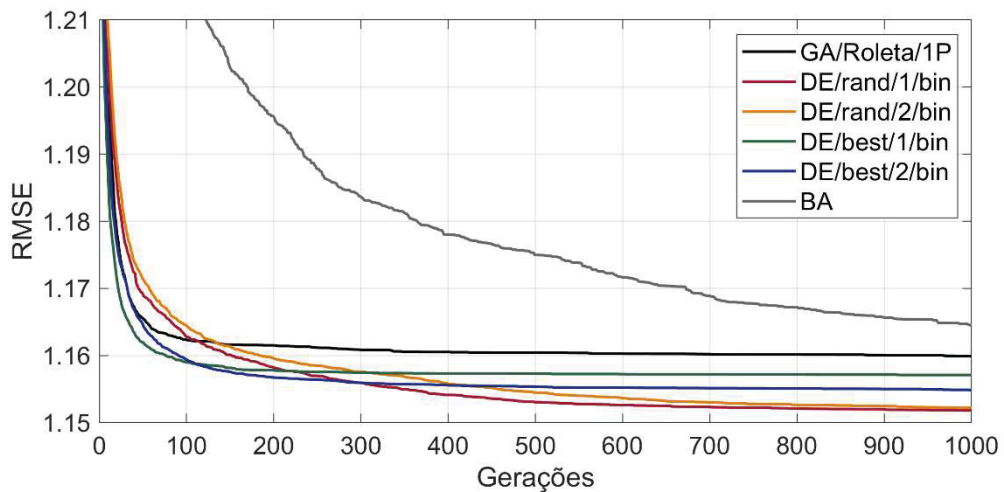
Observando a TABELA 18, é possível perceber que, embora o desempenho médio da abordagem GA/Roletta/1P tenha sido melhor, todas as abordagens foram capazes de alcançar o mesmo valor mínimo para o *RMSE*. Em outras palavras, as quatro abordagens localizaram o modelo que apresenta o *RMSE* da predição de um passo à frente igual a 1,1516. Este foi o modelo com menor valor de *RMSE* encontrado neste estudo de caso e está representado na equação (6.11). Para fins de nomenclatura, este modelo será chamado de AE01.

$$\begin{aligned}
y(k) = & 39,5771 + 0,9401y(k-1) - 1,5358y(k-2) \\
& + 0,0182y^2(k-2) + 0,0141y(k-2)u(k-1) \\
& + 3,8882 \times 10^{-5}y^2(k-1)y(k-2) \\
& - 1,7213 \times 10^{-4}y^2(k-1)u(k-1) \\
& + 2,8022 \times 10^{-5}y(k-1)u^2(k-1) \\
& - 1,0396 \times 10^{-4}y^3(k-3)
\end{aligned} \tag{6.11}$$

Mais uma vez, para análise das demais técnicas aplicadas neste trabalho, será utilizada como referência a curva do GA que alcançou o melhor desempenho médio neste estudo de caso, ou seja, a curva da abordagem GA/Roleta/1P.

Na FIGURA 39 é possível observar o desempenho do BA e das quatro estratégias da DE, quando estes foram associados à codificação binária TF.

FIGURA 39 – Convergência média para BA e DE codificados com TF (Estudo de caso 2)



FONTE: O autor (2020).

Observa-se, pela FIGURA 39, que a DE associada com a codificação binária TF alcançou valores médios menores que o GA com todas as estratégias, diferente do que ocorreu no estudo de caso da seção 6.2, onde somente duas das estratégias da DE conseguiram superar o GA. O BA codificado com TF, entretanto, novamente não atingiu desempenho melhor que o GA. Contudo, neste estudo de caso, a codificação binária TF fez com que tanto o BA quanto as quatro estratégias da DE

localizassem o modelo AE01. Os resultados estatísticos do *RMSE* estão apresentados na TABELA 19.

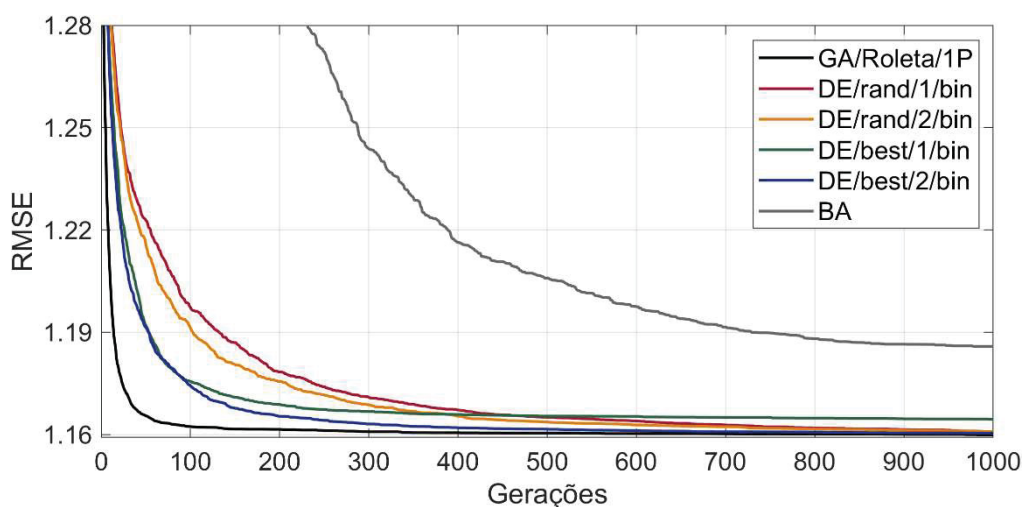
TABELA 19 – Valores de *RMSE* alcançados por BA e DE com codificação TF (Estudo de caso 2)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	<b>1,1516</b>	<b>1,1518</b>	<b>1,1547</b>	<b>0,000414</b>
DE/rand/2/bin	1,1516	1,1522	1,1549	0,000751
DE/best/1/bin	1,1516	1,1571	1,1672	0,003840
DE/best/2/bin	1,1516	1,1549	1,1631	0,003048
BA	1,1516	1,1645	1,2329	0,010704

FONTE: O autor (2020).

Quando utilizada a codificação binária GVP, os desempenhos alcançados pelos algoritmos BA e DE são os que estão apresentados na FIGURA 40. Neste caso, as quatro estratégias da DE atingiram desempenho médio bem próximo ao da abordagem GA/Roleta/1P, porém com valores de *RMSE* levemente maiores. O BA, por sua vez teve um desempenho baixo comparado ao da abordagem GA/Roleta/1P, apresentando valor médio de *RMSE* bem maior além de uma convergência mais lenta. Os valores estatísticos do *RMSE* obtidos com a codificação GVP estão listados na TABELA 20.

FIGURA 40 – Convergência média para BA e DE codificados com GVP (Estudo de caso 2)



FONTE: O autor (2020).

TABELA 20 – Valores de RMSE alcançados por BA e DE com codificação GVP (Estudo de caso 2)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	1,1586	1,1608	1,1654	0,001882
DE/rand/2/bin	1,1586	1,1609	1,1654	0,001787
DE/best/1/bin	1,1586	1,1645	1,1915	0,005992
DE/best/2/bin	<b>1,1586</b>	<b>1,1604</b>	<b>1,1686</b>	<b>0,002314</b>
BA	1,1604	1,1859	1,2096	0,012742

FONTE: O autor (2020).

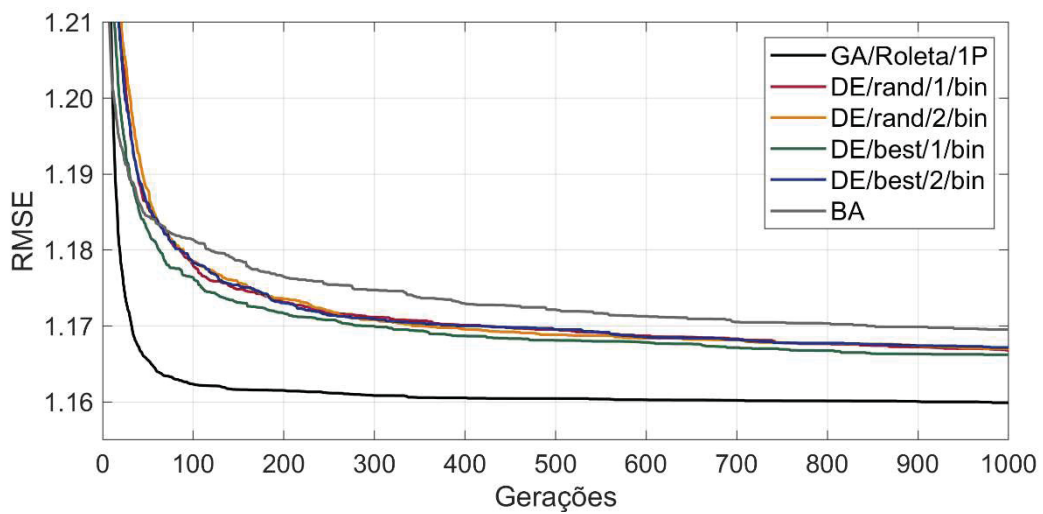
Pela TABELA 20 verifica-se que nenhuma das quatro estratégias da DE conseguiu localizar o modelo AE01, com a codificação GVP. Entretanto, todas chegaram ao mesmo modelo, com  $RMSE = 1,1586$ . Este modelo será denominado de AE02 e está apresentado na equação (6.12). O BA também não conseguiu localizar o modelo AE01 com a codificação GVP, chegando a um modelo com  $RMSE = 1,1604$ , o qual será denominado de AE03 e está apresentado na equação (6.13).

$$\begin{aligned}
y(k) = & 29,5713 - 0,2690y(k-2) + 0,5293u(k-1) \\
& + 0,0167y^2(k-1) - 0,0085y(k-2)u(k-1) \\
& - 8,9706 \times 10^{-5}y^3(k-1) \\
& - 0,0022y^2(k-1)u(k-1) \\
& + 0,0041y(k-1)y(k-2)u(k-1) \\
& + 0,0010y(k-1)u(k-1)u(k-2) \\
& - 0,0018y^2(k-2)u(k-1) \\
& - 0,0011y(k-2)u(k-1)u(k-2) \\
& - 5,0761 \times 10^{-5}u(k-1)u^2(k-2) \\
& + 7,6703 \times 10^{-5}u^3(k-2)
\end{aligned} \tag{6.12}$$

$$\begin{aligned}
y(k) = & 23,9844 + 0,3302u(k-1) + 0,0183y^2(k-1) \\
& - 0,0059y^2(k-2) - 0,0061u(k-1)u(k-2) \\
& - 0,0012y^2(k-1)y(k-2) \\
& + 0,0020y(k-1)y^2(k-2) \\
& - 6,1451 \times 10^{-5}y(k-1)u(k-1)u(k-2) \\
& + 7,0411 \times 10^{-4}y(k-1)u^2(k-2) \\
& - 8,4984 \times 10^{-4}y^3(k-2) \\
& + 3,9436 \times 10^{-5}y^2(k-2)u(k-1) \\
& - 6,7038 \times 10^{-4}y(k-2)u^2(k-2) \\
& + 1,2800 \times 10^{-5}u^3(k-2)
\end{aligned} \tag{6.13}$$

Com a codificação binária AM, a DE e o BA não foram capazes de alcançar resultados médios melhores que o da abordagem GA/Roleta/1P, como pode ser visto na FIGURA 41. Os valores estatísticos do *RMSE* obtidos com estas técnicas estão listados na TABELA 21.

FIGURA 41 – Convergência média para BA e DE codificados com AM (Estudo de caso 2)



FONTE: O autor (2020).



TABELA 21 – Valores de RMSE alcançados por BA e DE com codificação AM (Estudo de caso 2)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	1,1540	1,1668	1,1734	0,004349
DE/rand/2/bin	1,1519	1,1670	1,1745	0,004394
DE/best/1/bin	<b>1,1539</b>	<b>1,1662</b>	<b>1,1850</b>	<b>0,006209</b>
DE/best/2/bin	1,1560	1,1672	1,1854	0,004990
BA	1,1520	1,1695	1,2047	0,007754

FONTE: O autor (2020).

Observa-se, ainda, pela TABELA 21 que nenhuma das técnicas foi capaz de localizar o modelo AE01, com a codificação AM. Além disso, cada técnica chegou a um modelo diferente. Os melhores modelos localizados pelas estratégias DE/rand/1/bin, DE/rand/2/bin, DE/best/1/bin e DE/best/2/bin, receberão as nomenclaturas AE04, AE05, AE06 e AE07, respectivamente, e estarão apresentados nas equações (6.14) a (6.17). O melhor modelo localizado pelo BA com a codificação AM está apresentado na equação (6.18) e será denominado de AE08.

$$\begin{aligned}
 y(k) = & 33,9978 - 0,3278y(k-2) + 0,0139y^2(k-1) \\
 & + 0,0143y(k-2)u(k-1) \\
 & - 4,1742 \times 10^{-5}y^3(k-1) \\
 & - 1,7505 \times 10^{-4}y^2(k-1)u(k-1) \\
 & + 2,8783 \times 10^{-5}y(k-1)u^2(k-1)
 \end{aligned} \tag{6.14}$$

$$\begin{aligned}
 y(k) = & 53,7555 - 1,2390y(k-2) + 0,0279y(k-1)y(k-2) \\
 & + 0,0142y(k-2)u(k-1) \\
 & - 6,1822 \times 10^{-5}y^2(k-1)y(k-2) \\
 & - 1,7639 \times 10^{-4}y^2(k-1)u(k-1) \\
 & + 3,1386 \times 10^{-5}y(k-1)u^2(k-1) \\
 & - 5,0676 \times 10^{-5}y^3(k-2)
 \end{aligned} \tag{6.15}$$

$$\begin{aligned}
y(k) = & 49,5433 - 1,0599y(k-2) + 0,0253y(k-1)y(k-2) \\
& + 0,0143y(k-2)u(k-1) \\
& + 1,6944 \times 10^{-5}y^2(k-1)y(k-2) \\
& - 1,7294 \times 10^{-4}y^2(k-1)u(k-1) \\
& - 1,1850 \times 10^{-4}y(k-1)y^2(k-2) \\
& + 2,6586 \times 10^{-5}y(k-2)u^2(k-1)
\end{aligned} \tag{6.16}$$

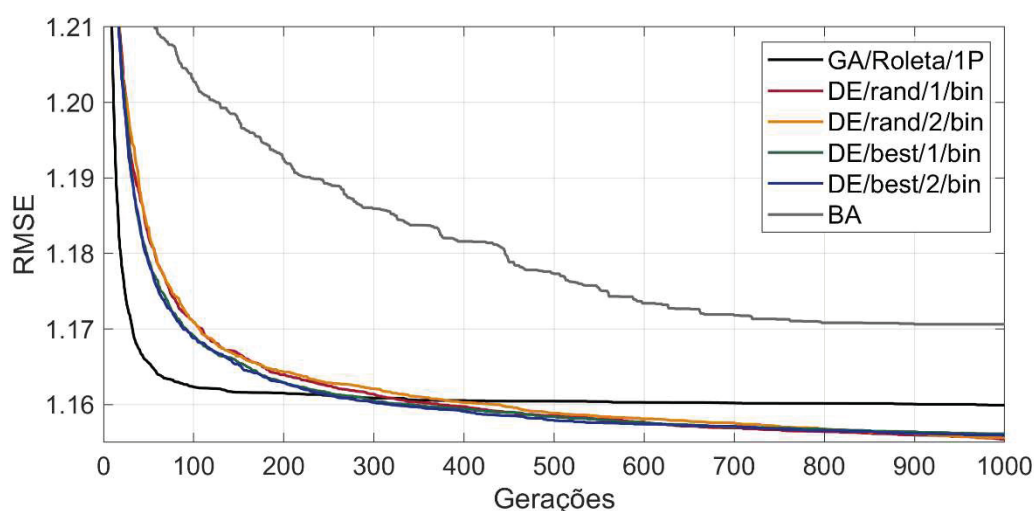
$$\begin{aligned}
y(k) = & 41,5672 + 1,1007y(k-1) - 1,8123y(k-2) \\
& + 0,0207y^2(k-2) + 0,0131y(k-2)u(k-1) \\
& - 1,5426 \times 10^{-4}y^2(k-1)u(k-1) \\
& + 2,9425 \times 10^{-5}y(k-1)y^2(k-2) \\
& + 2,3784 \times 10^{-5}y(k-1)u^2(k-1) \\
& - 1,1291 \times 10^{-4}y^3(k-2)
\end{aligned} \tag{6.17}$$

$$\begin{aligned}
y(k) = & 54,0017 - 1,2598y(k-2) + 0,0285y(k-1)y(k-2) \\
& + 0,0139y(k-2)u(k-1) \\
& - 7,1025 \times 10^{-5}y^2(k-1)y(k-2) \\
& - 1,5722 \times 10^{-4}y^2(k-1)u(k-1) \\
& - 4,8364 \times 10^{-5}y^3(k-2) \\
& + 2,2313 \times 10^{-5}u^3(k-1)
\end{aligned} \tag{6.18}$$

Os desempenhos da DE e do BA, ao se aplicar a codificação binária MAM, estão apresentados na FIGURA 42. Observa-se que os desempenhos das quatro estratégias da DE melhoram consideravelmente, em comparação ao que ocorreu quando foi aplicada a codificação AM, sendo que a média de redução do *RMSE* médio foi de 1%. Como foi visto na FIGURA 41, com a codificação AM, todas as quatro estratégias da DE convergiram para um *RMSE* médio maior que o da abordagem GA/Roleta/1P. Já utilizando a codificação MAM, as quatro estratégias alcançaram desempenhos médios melhores que o da abordagem GA/Roleta/1P, como pode ser

visto na FIGURA 42. O BA, entretanto, praticamente não teve alteração de desempenho ao mudar da codificação AM para a codificação MAM. Os valores estatísticos do  $RMSE$  para este caso podem ser verificados na TABELA 22.

FIGURA 42 – Convergência média para BA e DE codificados com MAM (Estudo de caso 2)



FONTE: O autor (2020).

TABELA 22 – Valores de RMSE alcançados por BA e DE com codificação MAM (Estudo de caso 2)

Estratégia	Mínimo	Médio	Máximo	Desvio Padrão
DE/rand/1/bin	<b>1,1517</b>	<b>1,1554</b>	<b>1,1605</b>	<b>0,002554</b>
DE/rand/2/bin	1,1516	1,1556	1,1603	0,002555
DE/best/1/bin	1,1516	1,1561	1,1619	0,002780
DE/best/2/bin	1,1516	1,1559	1,1624	0,002657
BA	1,1599	1,1706	1,1798	0,004169

FONTE: O autor (2020).

Outro ponto de destaque para a codificação MAM neste estudo de caso, é que, com ela quase todas as estratégias da DE passaram a localizar o modelo AE01, com exceção da estratégia DE/rand/1/bin, a qual localizou um modelo com  $RMSE = 1,1517$ , o segundo menor deste estudo de caso. Este modelo está apresentado na equação (6.19) e será denominado de AE09. Com a codificação MAM, o BA também não localizou o modelo AE01, e chegou ao modelo apresentado na equação (6.20), denominado de AE10.

$$\begin{aligned}
y(k) = & 56,6934 - 1,4111y(k-2) + 0,0314y(k-1)y(k-2) \\
& + 0,0131y(k-2)u(k-1) \\
& - 9,4006 \times 10^{-5}y^2(k-1)y(k-2) \\
& - 1,5433 \times 10^{-4}y^2(k-1)u(k-1) \\
& - 4,4315 \times 10^{-5}y^3(k-2) \\
& + 2,3843 \times 10^{-5}y(k-2)u^2(k-1)
\end{aligned} \tag{6.19}$$

$$\begin{aligned}
y(k) = & 42,7681 - 0,8760y(k-2) + 0,0267y^2(k-1) \\
& + 0,0014y(k-1)u(k-2) + 0,0080y(k-2)u(k-1) \\
& - 0,0115u(k-1)u(k-2) - 4,4192 \times 10^{-4}y^3(k-1) \\
& + 6,9727 \times 10^{-4}y(k-1)y^2(k-2) \\
& + 6,1353 \times 10^{-4}y(k-1)u^2(k-2) \\
& - 4,0082 \times 10^{-4}y^3(k-2) \\
& - 3,8604 \times 10^{-5}y^2(k-2)u(k-2) \\
& - 5,4101 \times 10^{-4}y(k-2)u^2(k-2)
\end{aligned} \tag{6.20}$$

Além de analisar cada técnica através da convergência média, foi verificada também a taxa de sucesso para localizar um modelo de referência, conforme definido na equação (5.4). O modelo AE01 foi o que apresentou o menor *RMSE* neste estudo de caso, portanto o mesmo será considerado como modelo de referência para o cálculo da taxa de localização,  $L$ , de cada técnica. Os resultados podem ser observados na TABELA 23.

TABELA 23 – Taxa de localização das técnicas aplicadas na seleção de estrutura (Estudo de caso 2)

Codificação Binária	Estratégia	L	$L_{MÉDIO} \pm \text{desv. pad.}$
-	GA/Roleta/1P	6%	5,25% $\pm$ 0,02
-	GA/Roleta/2P	7%	
-	GA/Torneio/1P	3%	
-	GA/Torneio/2P	5%	
TF	DE/rand/1/bin	55%	23,40% $\pm$ 0,21
	DE/rand/2/bin	19%	
	DE/best/1/bin	11%	
	DE/best/2/bin	31%	
	BA	1%	
GVP	DE/rand/1/bin	0%	0% $\pm$ 0,00
	DE/rand/2/bin	0%	
	DE/best/1/bin	0%	
	DE/best/2/bin	0%	
	BA	0%	
AM	DE/rand/1/bin	0%	0% $\pm$ 0,00
	DE/rand/2/bin	0%	
	DE/best/1/bin	0%	
	DE/best/2/bin	0%	
	BA	0%	
MAM	DE/rand/1/bin	0%	1,80% $\pm$ 0,02
	DE/rand/2/bin	1%	
	DE/best/1/bin	4%	
	DE/best/2/bin	4%	
	BA	0%	

FONTE: O autor (2020).

Analizando a TABELA 23, observa-se que, diferente do estudo de caso anterior (seção 6.2), em termos gerais as técnicas atingiram valores reduzidos de taxa de localização. Isto pode ser considerado uma consequência do tamanho do espaço de busca neste estudo de caso, que é muito maior do que no estudo de caso anterior, o que torna mais árdua a tarefa de localizar um determinado modelo.

É possível verificar também que, em termos gerais, as maiores taxas de localização foram alcançadas pela DE em conjunto com a codificação TF. As codificações GVP e AM, por sua vez, não possibilitaram nenhuma das metaheurísticas localizarem o modelo de referência, levando a uma taxa de localização de 0%. Entretanto, percebe-se uma melhora nestes resultados ao ser aplicada a codificação MAM, que levou as estratégias DE/rand/1/bin, DE/best/1/bin e DE/best/2/bin a localizarem o modelo de referência. Em termos de valores médios, a codificação TF foi a que apresentou o melhor desempenho, superando inclusive o GA.

O custo computacional também foi verificado para cada técnica através do tempo de simulação medido em segundos, com seus resultados indicados na TABELA 24.

TABELA 24 – Custo computacional de cada técnica utilizada (Estudo de caso 2)

<b>Codificação Binária</b>	<b>Estratégia</b>	<b>Tempo (s) (média ± desv. pad.)</b>
-	GA/Roleta/1P	218 ± 16
-	GA/Roleta/2P	215 ± 13
-	GA/Torneio/1P	248 ± 17
-	GA/Torneio/2P	232 ± 16
TF	DE/rand/1/bin	206 ± 15
	DE/rand/2/bin	207 ± 14
	DE/best/1/bin	212 ± 17
	DE/best/2/bin	213 ± 16
	BA	194 ± 12
GVP	DE/rand/1/bin	223 ± 12
	DE/rand/2/bin	224 ± 13
	DE/best/1/bin	228 ± 15
	DE/best/2/bin	232 ± 15
	BA	184 ± 29
AM	DE/rand/1/bin	185 ± 26
	DE/rand/2/bin	187 ± 28
	DE/best/1/bin	183 ± 26
	DE/best/2/bin	185 ± 27
	BA	183 ± 6
MAM	DE/rand/1/bin	215 ± 7
	DE/rand/2/bin	216 ± 8
	DE/best/1/bin	218 ± 8
	DE/best/2/bin	218 ± 9
	BA	163 ± 21

FONTE: O autor (2020).

Observa-se da TABELA 24 que o tempo médio de simulação do GA e da DE associada às codificações TF, GVP e MAM, em geral são próximos, estando todos acima dos 200s. Já a codificação AM permitiu que a DE e o BA atingissem tempos menores de simulação, todos abaixo dos 190s, em média. O BA foi a técnica que chegou aos menores tempos médios com todas as codificações, alcançando com a codificação MAM o menor tempo médio deste estudo de caso.

### 6.3.3 Validação dos modelos obtidos

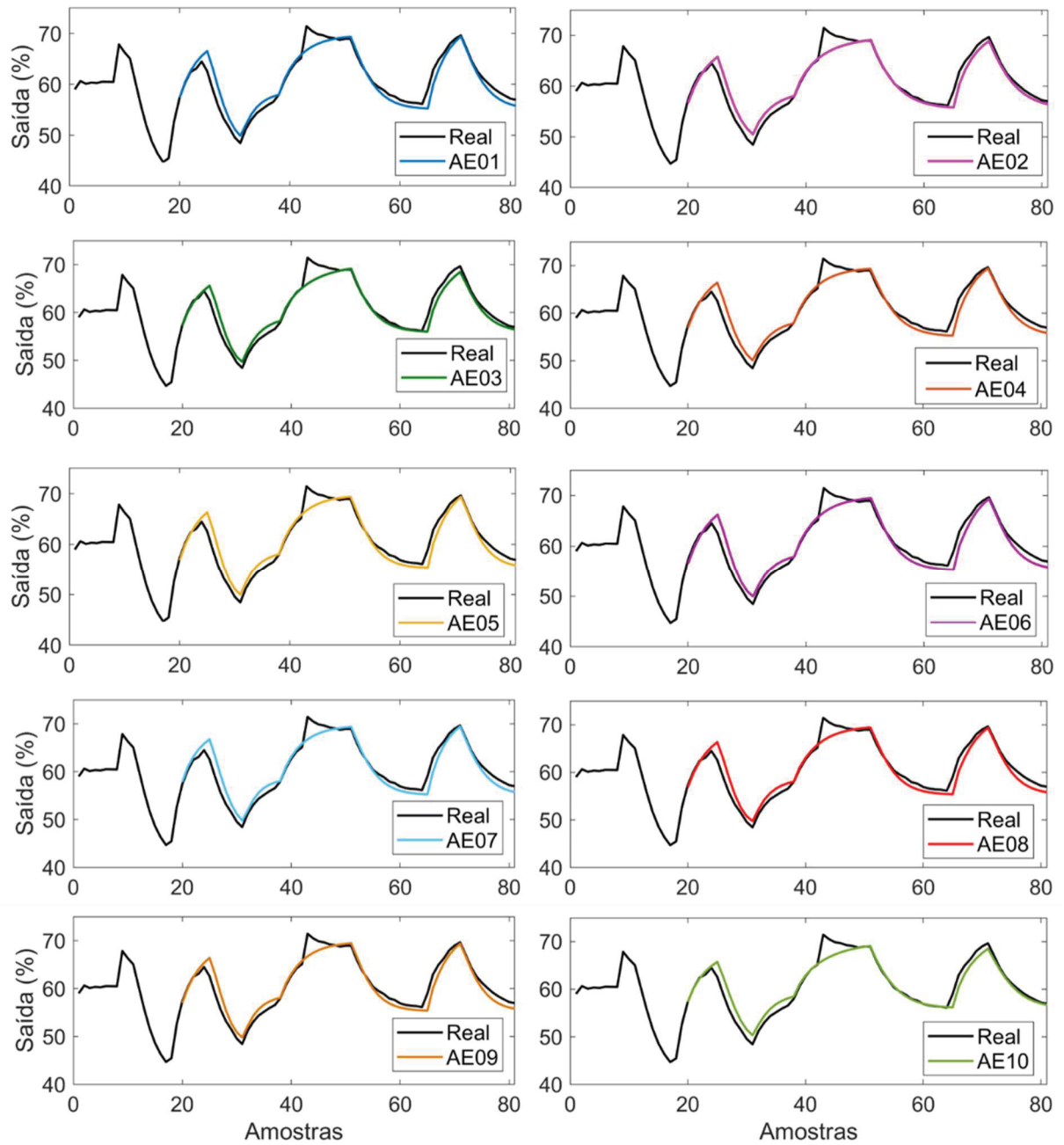
Nesta seção serão apresentadas as validações dos melhores modelos encontrados para o aquecedor elétrico, ou seja, os modelos AE01 a AE10. Os procedimentos utilizados serão simulação livre e análise de resíduos. Vale ressaltar que, ao aplicar os dados apresentados na FIGURA 36(b) para validar os modelos através de simulação livre, Rodrigues (1996) utilizou as primeiras 19 amostras para inicializar os modelos obtidos. Portanto, inicialmente, tal procedimento também será adotado nesta seção.

As simulações livres dos modelos selecionados encontram-se apresentadas na FIGURA 43, onde observa-se que todos os modelos se ajustam bem aos dados dinâmicos do sistema. Vale ressaltar que estes modelos foram obtidos a partir da massa de dados da FIGURA 36(a), a qual não apresenta as falhas identificadas nas amostras amostras  $k \approx 10$  e  $k \approx 43$ , verificadas na massa de dados da FIGURA 36(b). Desta forma, é possível observar na FIGURA 43 que os modelos não predizem essas falhas, o que indica uma boa consistência nos resultado.

Além das verificações visuais observadas nos gráficos da FIGURA 43, foram calculados os valores de  $RMSE_{SL}$  e  $R_{SL}^2$  sobre os dados de validação, os quais podem ser observados na TABELA 25. Nesta tabela também foram incluídos os mesmos índices calculados sobre os dados de estimação.

Através da TABELA 25 verifica-se que o modelo AE03 apresenta melhores valores para os índices  $RMSE_{SL}$  e  $R_{SL}^2$ , calculados sobre os dados de validação. Portanto, segundo estes critérios, o mesmo pode ser considerado o modelo que melhor se ajusta aos dados dinâmicos do sistema ao utilizar a simulação livre. Entretanto, este é um dos modelos com mais termos encontrados neste estudo de caso (10 regressores mais a constante), o que pode levar à escolha de algum dos demais modelos obtidos, caso haja necessidade de um modelo menos complexo e seja permitido uma certa tolerância nos valores dos índices  $RMSE_{SL}$  e  $R_{SL}^2$ .

FIGURA 43 – Simulação livre dos modelos AE01 a AE10



FONTE: O autor (2020).



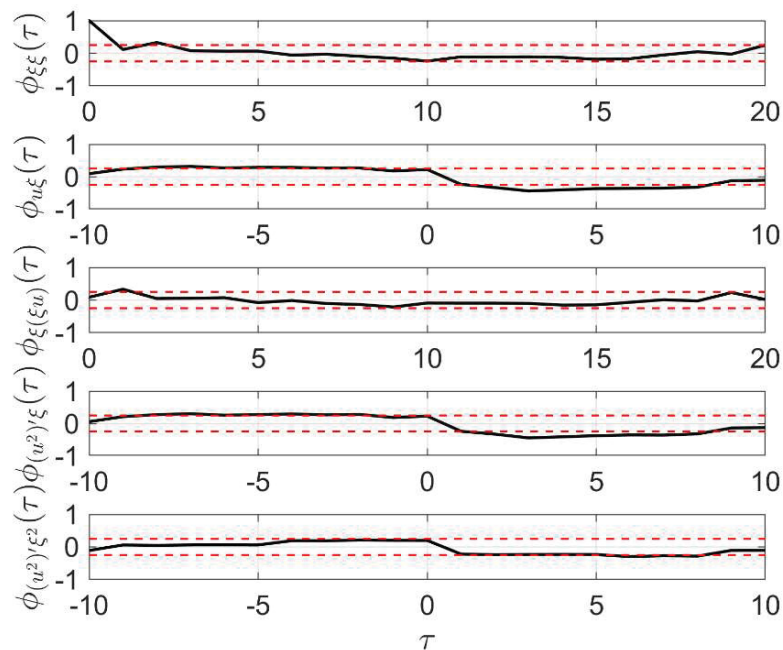
TABELA 25 – Índices de desempenho para os modelos AE01 a AE10

Modelo	Estimação		Validação	
	$RMSE_{SL}$	$R^2_{SL}$	$RMSE_{SL}$	$R^2_{SL}$
AE01	0,8405	97,10%	1,5877	92,21%
AE02	0,9504	96,30%	1,4969	93,08%
AE03	<b>0,9319</b>	<b>96,44%</b>	<b>1,4898</b>	<b>93,14%</b>
AE04	0,8436	97,08%	1,5697	92,39%
AE05	0,8440	97,08%	1,5617	92,46%
AE06	0,8378	97,12%	1,5464	92,61%
AE07	0,8439	97,08%	1,6210	91,88%
AE08	0,8501	97,04%	1,5361	92,71%
AE09	0,8454	97,07%	1,5629	92,45%
AE10	0,9561	96,25%	1,5603	92,48%

FONTE: O autor (2020).

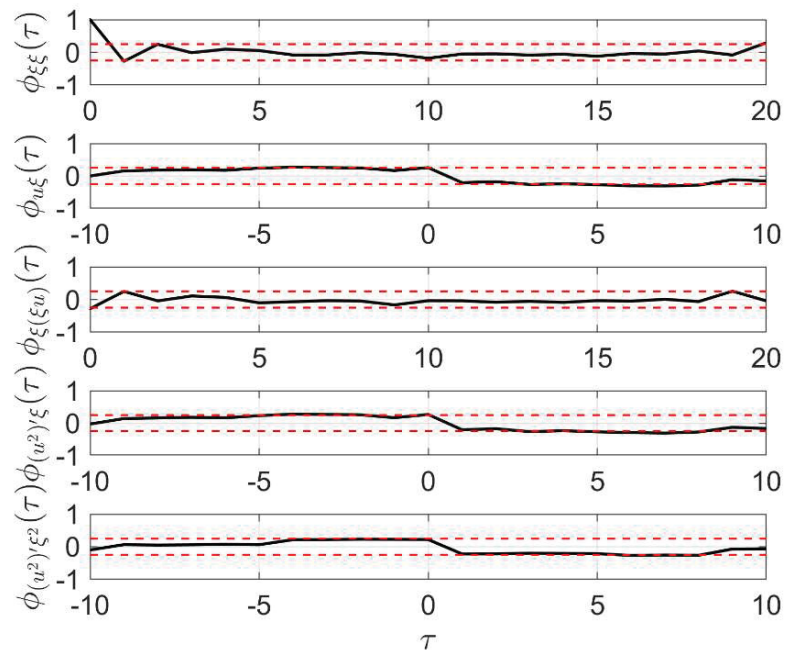
O próximo método aplicado para avaliar os modelos é a validação estatística, onde será possível analisar o comportamento dos resíduos e verificar se estes são aleatórios, assemelhando-se a um ruído branco. Nas FIGURA 44 a 53 é possível observar as análises de resíduos para os modelos AE01 a AE10, respectivamente.

FIGURA 44 – Validação estatística do modelo AE01.



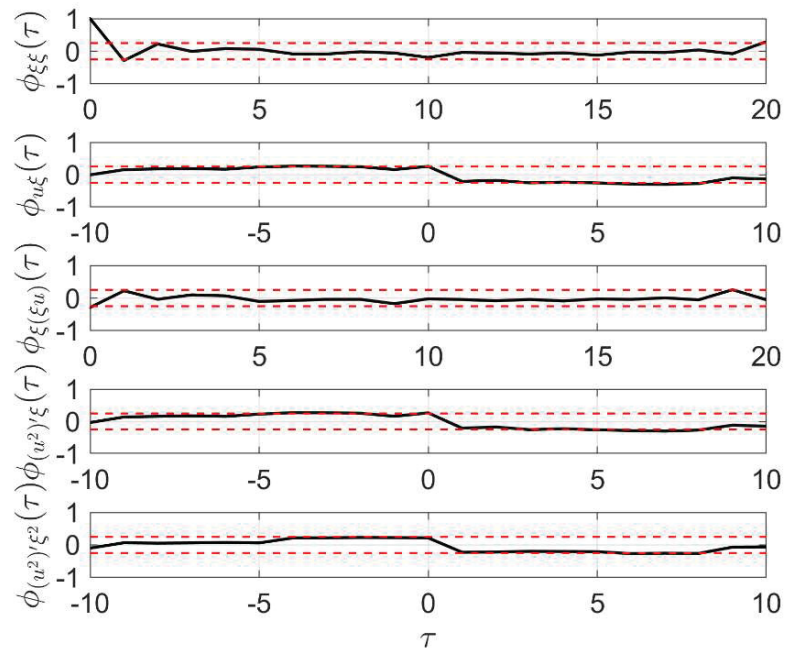
FONTE: O autor (2020).

FIGURA 45 – Validação estatística do modelo AE02.



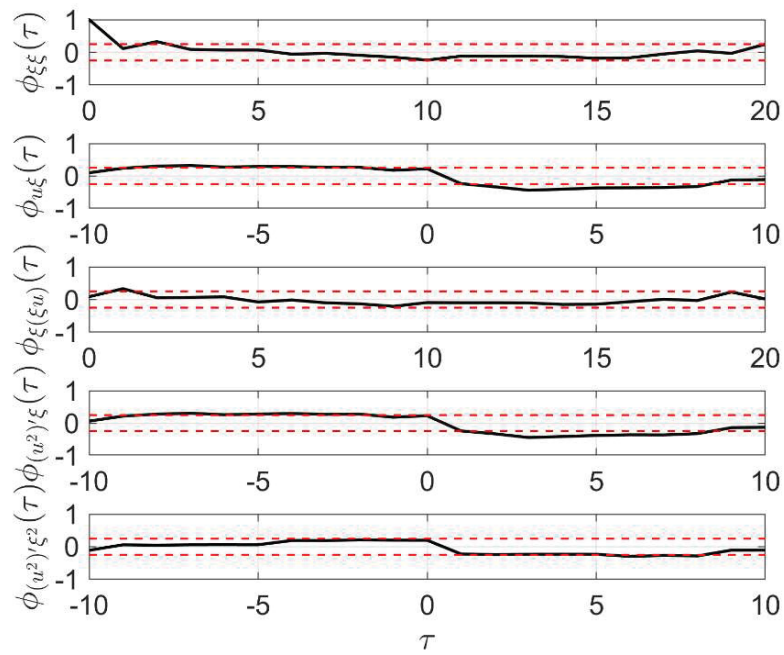
FONTE: O autor (2020).

FIGURA 46 – Validação estatística do modelo AE03.



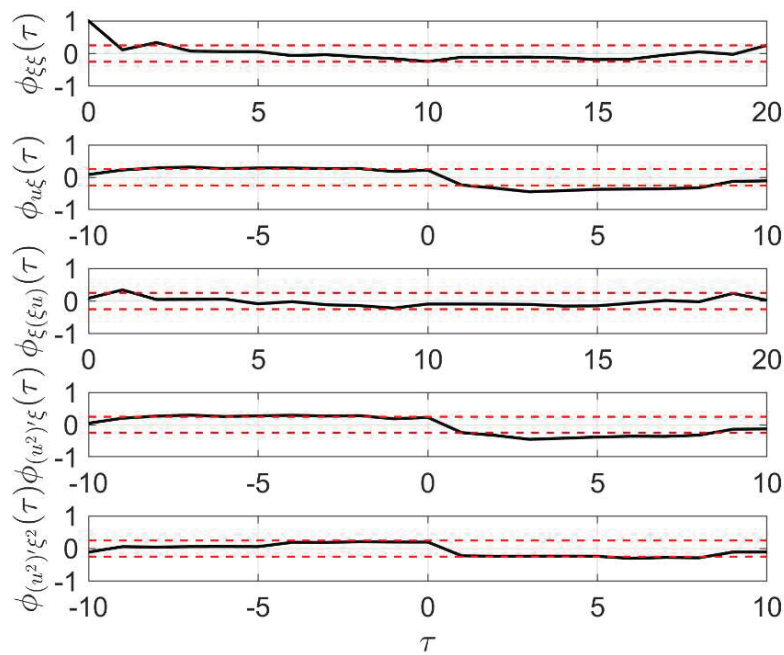
FONTE: O autor (2020).

FIGURA 47 – Validação estatística do modelo AE04.



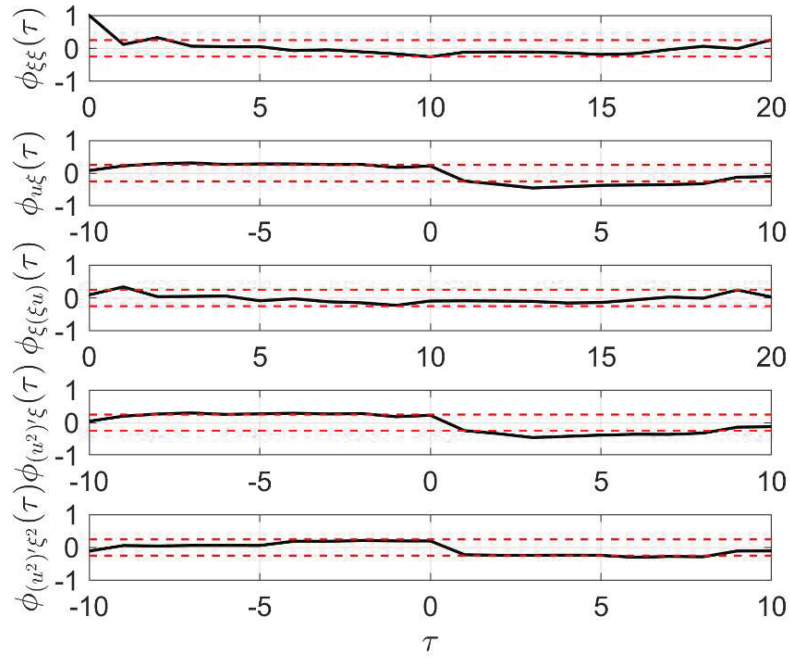
FONTE: O autor (2020).

FIGURA 48 – Validação estatística do modelo AE05.



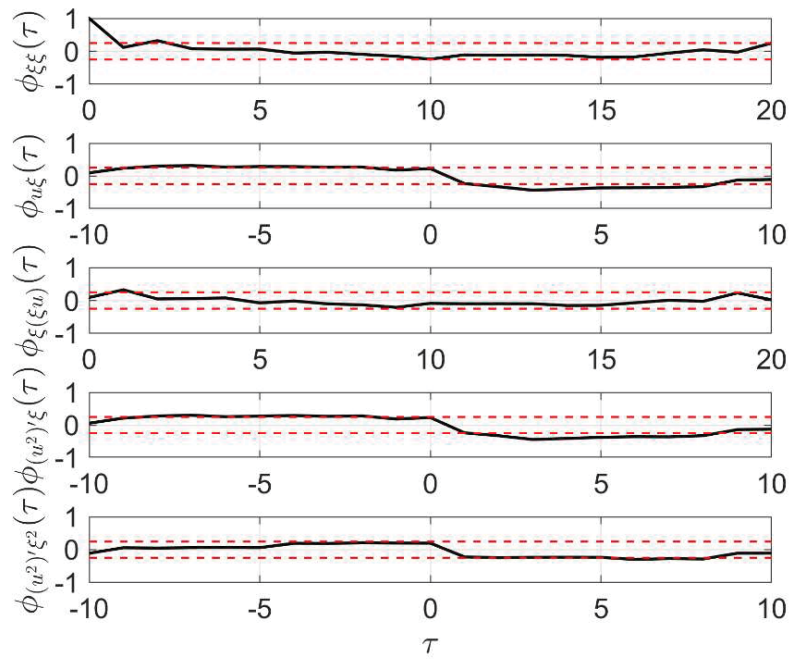
FONTE: O autor (2020).

FIGURA 49 – Validação estatística do modelo AE06.



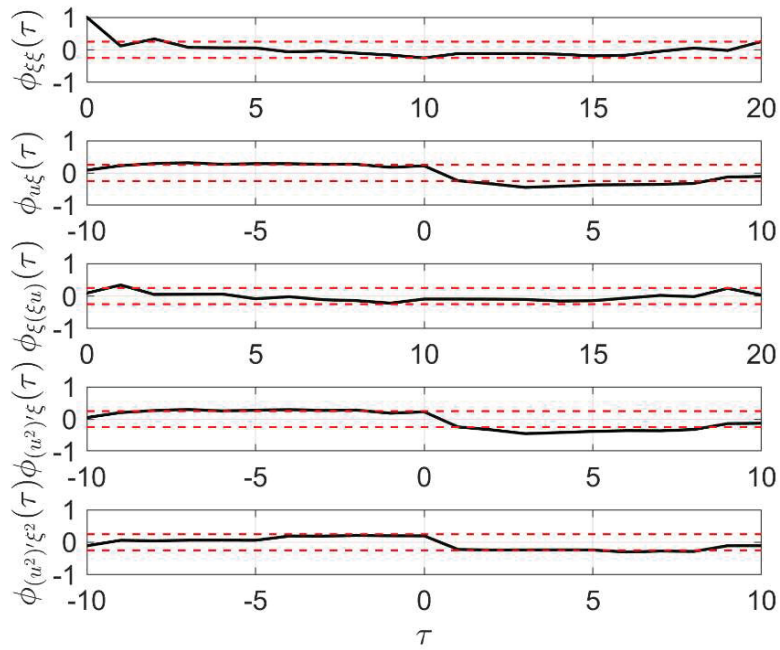
FONTE: O autor (2020).

FIGURA 50 – Validação estatística do modelo AE07.



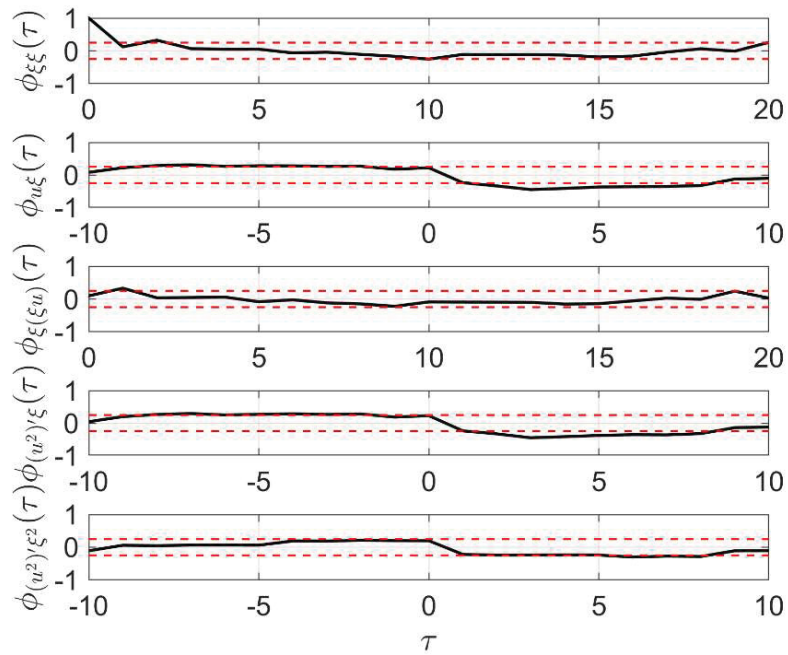
FONTE: O autor (2020).

FIGURA 51 – Validação estatística do modelo AE08.



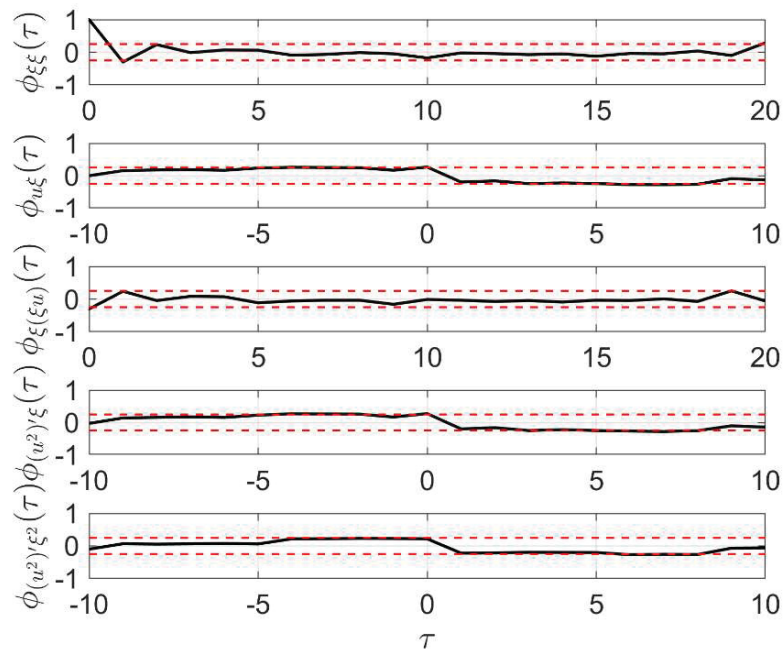
FONTE: O autor (2020).

FIGURA 52 – Validação estatística do modelo AE09.



FONTE: O autor (2020).

FIGURA 53 – Validação estatística do modelo AE10.



FONTE: O autor (2020).

Analisando as FIGURAS 44 a 53, percebe-se que, no geral as funções de autocorrelação e correlação cruzada encontram-se dentro do intervalo de confiança de 95% (linhas vermelhas tracejadas), para a maioria dos modelos, sugerindo o comportamento aleatório dos resíduos. Em alguns modelos, como por exemplo o AE01, verifica-se que a correlação cruzada linear ( $\phi_{u\xi}$ ) sai um pouco da faixa delimitada pelo intervalo de confiança em  $\tau = 4$ , o que não compromete a qualidade dos modelos.

Uma outra verificação realizada foi em relação à complexidade dos modelos e presença de determinados regressores em suas estruturas. As informações referentes à quantidade de termos em cada modelo e ao número de ocorrências de cada termos nos modelos AE01 a AE10 foram consolidadas e estão apresentadas na FIGURA 54. Observa-se que o modelo com menos termos é o modelo AE04 (7 termos) e o modelo com mais termos é o modelo AE02 (13 termos). Também é possível verificar o grau de importância dos termos  $y(k-2)$ ,  $y(k-2)u(k-1)$ , assim como a constante, os quais ocorreram em todos os modelos. Outros regressores que podem ser considerados relevantes são os que ocorreram em mais



da metade do modelos, que são os regressores  $y^2(k-2)u(k-1)$  (8 ocorrências) e  $y^3(k-2)$  (7 ocorrências). Além disso, pela FIGURA 54 também é possível verificar diversos regressores que não apresentaram relevância, segundo as técnicas utilizadas, uma vez que não apareceram em nenhum dos modelos identificados (0 ocorrências).

FIGURA 54 – Análise dos regressores dos modelos AE01 a AE10.

		AE01	AE02	AE03	AE04	AE05	AE06	AE07	AE08	AE09	AE10	OCORRÊNCIAS DOS TERMOS
	QUANTIDADE DE TERMOS	9	13	11	7	8	8	9	8	8	12	
1ª	1	x	x	x	x	x	x	x	x	x	x	10
2ª	$y(k-1)$	x						x				2
3ª	$y(k-2)$	x	x	x	x	x	x	x	x	x	x	10
4ª	$u(k-1)$		x									1
5ª	$u(k-2)$											0
6ª	$y^2(k-1)$		x	x	x						x	4
7ª	$y(k-1)y(k-2)$					x	x		x	x		4
8ª	$y(k-1)u(k-1)$											0
9ª	$y(k-1)u(k-2)$										x	1
10ª	$y^2(k-2)$	x						x				2
11ª	$y(k-2)u(k-1)$	x	x	x	x	x	x	x	x	x	x	10
12ª	$y(k-2)u(k-2)$											0
13ª	$u^2(k-1)$											0
14ª	$u(k-1)u(k-2)$			x							x	2
15ª	$u^2(k-2)$											0
16ª	$y^3(k-1)$		x	x	x						x	4
17ª	$y^2(k-1)y(k-2)$	x				x	x		x	x		5
18ª	$y^2(k-1)u(k-1)$	x	x		x	x	x	x	x	x		8
19ª	$y^2(k-1)u(k-2)$											0
20ª	$y(k-1)y^2(k-2)$			x			x	x			x	4
21ª	$y(k-1)y(k-2)u(k-1)$		x									1
22ª	$y(k-1)y(k-2)u(k-2)$											0
23ª	$y(k-1)u^2(k-1)$	x			x	x		x				4
24ª	$y(k-1)u(k-1)u(k-2)$		x									1
25ª	$y(k-1)u^2(k-2)$			x							x	2
26ª	$y^3(k-2)$	x		x		x		x	x	x	x	7
27ª	$y^2(k-2)u(k-1)$		x									1
28ª	$y^2(k-2)u(k-2)$										x	1
29ª	$y(k-2)u^2(k-1)$						x			x		2
30ª	$y(k-2)u(k-1)u(k-2)$		x									1
31ª	$y(k-2)u^2(k-2)$			x							x	2
32ª	$u^3(k-1)$								x			1
33ª	$u^2(k-1)u(k-2)$											0
34ª	$u(k-1)u^2(k-2)$		x									1
35ª	$u^3(k-2)$		x	x								2

FONTE: O autor (2020).

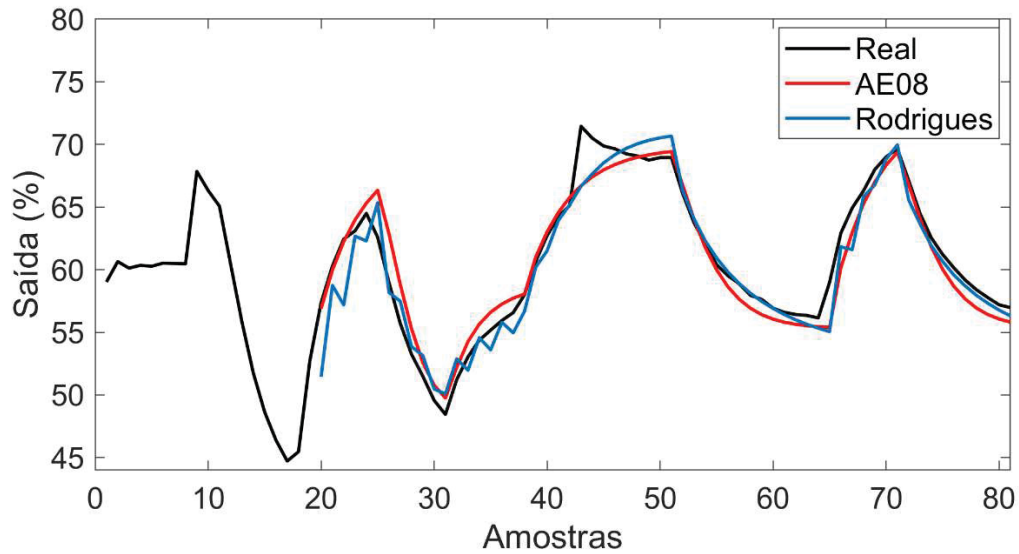
#### 6.3.4 Comparações com a literatura

No trabalho de Rodrigues (1996), foram estimados modelos para o aquecedor elétrico utilizando MQ ortogonais associado ao ERR. Um dos melhores modelos indicados no trabalho é o que está apresentado na equação (6.21).

$$\begin{aligned}
 y(k) = & -0,96659y(k-1) + 1,81300y(k-2) \\
 & - 0,01888u(k-2) + 0,74820u(k-1) \\
 & - 5,65530 \times 10^{-5}y^2(k-1)u(k-1) \\
 & - 7,4452 \times 10^{-5}u^3(k-1) \\
 & - 1,15340 \times 10^{-4}y^3(k-2) \\
 & + 1,16990 \times 10^{-4}y^3(k-1)
 \end{aligned} \tag{6.21}$$

Para realizar uma comparação com o modelo de Rodrigues (1996), por meio de simulação livre e cálculo dos índices  $RMSE_{SL}$  e  $R_{SL}^2$ , optou-se por utilizar um modelo com a mesma quantidade de termos (8 termos). Dentre os modelos que atendiam este critério, foi escolhido o modelo AE08 por apresentar o menor valor de  $RMSE_{SL}$ , calculado sobre os dados de validação. Como discutido anteriormente, foi seguido o mesmo procedimento realizado por Rodrigues (1996), ao utilizar as primeiras 19 amostras dos dados de validação para inicializar os modelos. Ou seja, o modelo começa o processo de predição livre a partir da amostra  $k = 20$ . Os resultados desta comparação podem ser observados na FIGURA 55 e na TABELA 26. Nesta comparação, somente os dados de validação foram utilizados.



FIGURA 55 – Simulação livre dos modelos AE08 e Rodrigues (início:  $k = 20$ )

FONTE: O autor (2020).

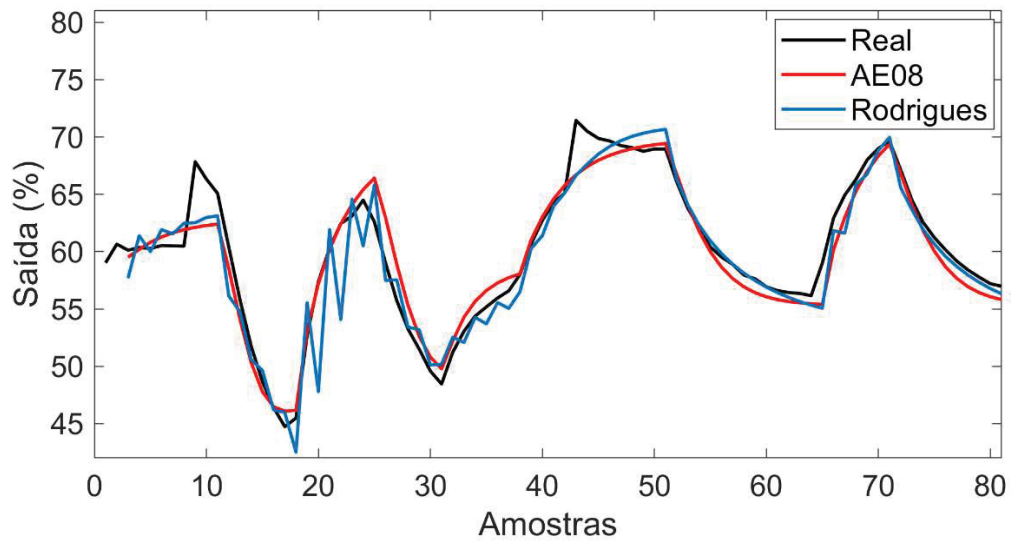
TABELA 26 – Índices de desempenho para os modelos AE08 e Rodrigues (início:  $k = 20$ )

Modelo	$RMSE_{SL}$	$R_{SL}^2$
AE08	1,5361	92,71%
Rodrigues	1,7034	91,03%

FONTE: O autor (2020).

Analizando a FIGURA 55, observa-se que o modelo AE08 segue os dados reais de uma maneira mais suave, ajustando de uma forma mais coerente os dados dinâmicos do sistema, enquanto o modelo de Rodrigues (1996) apresenta uma série de oscilações em algumas regiões. Além disso, pela TABELA 26, verificam-se os melhores valores dos índices de desempenho do modelo AE08 em relação ao modelo de Rodrigues (1996). O modelo AE08 apresenta o  $RMSE_{SL}$  9,82% menor e o  $R_{SL}^2$  1,68% maior, em relação ao modelo de Rodrigues.

Uma outra comparação foi realizada, desta vez utilizando os dados de validação de maneira integral, ou seja, como o atraso no sinal de saída é  $n_y = 2$ , apenas as duas primeiras amostras foram utilizadas para inicializar os modelos, fazendo com que a predição livre iniciasse na amostra  $k = 3$ . Os resultados desta segunda comparação podem ser observados na FIGURA 56 e na TABELA 27.

FIGURA 56 – Simulação livre dos modelos AE08 e Rodrigues (início:  $k = 3$ )

FONTE: O autor (2020).

TABELA 27 – Índices de desempenho para os modelos AE08 e Rodrigues (início:  $k = 3$ )

Modelo	$RMSE_{SL}$	$R^2_{SL}$
AE08	1,6718	92,87%
Rodrigues	2,2332	87,28%

FONTE: O autor (2020).

Observando a FIGURA 56, verifica-se que o modelo AE08 mantém um bom ajuste aos dados dinâmicos do sistema, mesmo inicializando apenas com as duas primeiras amostras. O formato dos dados reais é seguido de maneira suave e as falhas apresentadas nas amostras  $k \approx 10$  e  $k \approx 43$ , continuam não sendo preditas. O modelo de Rodrigues (1996) também não prediz estas falhas, porém observa-se que o ajuste aos dados piora quando o modelo é inicializado com as primeiras duas amostras. As oscilações aumentam e os índices de desempenho pioram consideravelmente. Neste segundo cenário, modelo AE08 apresenta o  $RMSE_{SL}$  25,14% menor e o  $R^2_{SL}$  5,59% maior, em relação ao modelo de Rodrigues. Com isto, percebe-se que o modelo AE08 não só possui um melhor comportamento em termos dos índices de desempenho, como também apresenta uma maior versatilidade, uma vez que faz um ajuste aos dados dinâmicos independente do ponto de inicialização.

## 7 CONSIDERAÇÕES FINAIS

Este trabalho apresentou um estudo comparativo entre metaheurísticas aplicadas ao problema de seleção de estrutura de modelos NARX polinomiais. Dois tipos de metaheurísticas foram utilizados: as que originalmente manipulam soluções binárias e aquelas projetadas para trabalhar com valores contínuos e, por isso, precisam de uma codificação para manipular soluções no espaço de busca binário.

A metaheurística de soluções binárias aplicada foi o Algoritmo Genético (GA) e as metaheurísticas de soluções contínuas utilizadas foram a Evolução Diferencial (DE) e o Algoritmo do Morcego (BA). Para implementar as versões binárias da DE e do BA, foram aplicadas as codificações por Função de Transferência (TF), Prioridade de Maior Valor (GVP), Modulação em Ângulo (AM) e Modulação em Ângulo Modificada (MAM), sendo esta última uma proposta deste trabalho.

Foram realizados dois estudos de caso utilizando dados de sistemas disponíveis na literatura. O primeiro aborda um conversor *buck* e o segundo trata de um aquecedor elétrico. Por ser uma técnica consagrada no problema de seleção de estrutura, o GA foi aplicado para obter modelos dos sistemas e também para servir de referência para as demais técnicas.

No primeiro estudo de caso, todas as abordagens do GA analisadas apresentaram bom desempenho, no sentido de que foram capazes de localizar o modelo considerado ótimo em mais da metade das simulações realizadas (52%). Já no segundo estudo de caso, embora o GA tenha localizado o modelo de referência com todas as suas abordagens, isto ocorreu numa quantidade bem menor de simulações (5%) reduzindo consideravelmente o seu desempenho.

Utilizando a codificação TF, observou-se um desempenho satisfatório da DE em termos de convergência média, uma vez que duas das quatro estratégias (DE/rand/1bin e DE/rand/2/bin) superaram o GA no primeiro estudo de caso e todas as quatro estratégias superaram o GA no segundo estudo de caso. Já o BA associado à codificação TF, não conseguiu atingir desempenho melhor que o GA em nenhum estudo de caso. No quesito localização do modelo de referência, a codificação TF levou a DE a atingir uma taxa média de 64% no primeiro estudo de caso e 29% no segundo. O BA, por sua vez, atingiu os percentuais de 0%, no primeiro estudo de caso, e 1% no segundo.

Ao aplicar a codificação GVP, o comportamento da DE, em termos de convergência média, se repetiu com as estratégias DE/rand/1bin e DE/rand/2/bin superando o GA no primeiro estudo de caso. Contudo, no segundo estudo de caso, o desempenho das quatro estratégias da DE apenas chegaram bem próximos ao do GA. Os resultados para o BA com a codificação GVP, foram semelhantes aos obtidos com a codificação TF, ou seja, os desempenhos ficaram bem abaixo do GA, tanto no primeiro quanto no segundo estudo de caso. Quanto à taxa de localização, no primeiro estudo de caso a DE atingiu um valor médio de 53% e o BA obteve 0% de localização do modelo de referência. Já no segundo estudo de caso, todas as técnicas tiveram 0% de localização com a codificação GVP.

Com a codificação AM, as quatro estratégias da DE e também o BA tiveram desempenhos piores que o do GA em termos de convergência, tanto no primeiro, quanto no segundo estudo de caso. Além disso, as taxas de localização média para a DE foram de 0,5% e 0%, no primeiro e segundo estudos de caso, respectivamente. Para o BA, em ambos os estudos de caso este resultado foi de 0%.

Aplicando-se a versão modificada da codificação AM (MAM), a qual é proposta neste trabalho, foi possível atingir melhorias significativas no desempenho da DE. No primeiro estudo de caso, três das quatro estratégias superaram a convergência do GA e uma delas atingiu desempenho próximo ao do GA. Já no segundo estudo de caso, todas as quatro estratégias tiveram desempenhos melhores que o GA, em termos de convergência. Ou seja, com a codificação MAM as quatro estratégias da DE atingiram desempenhos melhores do que aqueles obtidos com a codificação AM, em ambos os estudos de caso. Isso também pode ser verificado na taxa de localização média da DE, que foi de 0,5% para 73% no primeiro estudo de caso, e de 0% para 2% no segundo estudo de caso. Para o BA, a codificação MAM trouxe melhorias pouco significativas no primeiro estudo de caso, e nenhuma melhoria no segundo.

No quesito custo computacional, foi verificado o tempo médio de simulação utilizado por cada técnica. No primeiro estudo de caso, verificou-se um padrão de duração das simulações (valores médios entre 340s e 380s) para quase todas as técnicas, com exceção do BA, obteve tempos médios bastante divergentes para cada tipo de codificação. Já no segundo estudo de caso, o padrão do tempo médio de simulação obtido pela maioria das técnicas foi de valores acima de 200s. Entretanto,

a codificação AM destacou-se alcançando, em todos os cenários, tempos abaixo de 190s.

Em ambos os estudos de caso, os melhores modelos obtidos apresentaram bons resultados quando submetidos às validações do tipo simulação livre e análise de resíduos. Estes modelos também mostraram adequar-se melhor aos dados dinâmicos do sistema quando comparados com modelos da literatura.

Vale ressaltar que o estudo apresentado foi focado em obter modelos com o maior ajuste possível aos dados dinâmicos do sistema, não priorizando, num primeiro momento, outros tipos de características como complexidade do modelo e aderência à curva estática do sistema.

## 7.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

As possibilidades de continuidade deste trabalho envolvem os seguintes tópicos, mas não limitadas a estes:

- Replicar as mesmas análises utilizando outros tipos de modelos como, por exemplo, o modelo NARMAX;
- Adotar outros estimadores para determinar os valores dos parâmetros dos modelos;
- Aplicar os métodos apresentados para obtenção de modelos de outros sistemas;
- Utilizar outras metaheurísticas de espaço contínuo para reforçar as conclusões acerca das técnicas de codificação binária;
- Explorar outros métodos de codificação binária;
- Realizar a busca de modelos levando em conta também sua complexidade.

## REFERÊNCIAS

ABRASHOV, S.; MALTI, R.; MOREAU, X.; MOZE, M.; AIOUN, F.; GUILLEMARD, F. Optimal input design for continuous-time system identification. **Communications in Nonlinear Science and Numerical Simulation**, v. 60, p. 92–99, 2018.

AGUIRRE, L. A. Some Remarks on Structure Selection for Nonlinear Models. **International Journal of Bifurcation and Chaos**, v. 4, n. 6, p. 1707–1714, 20 dez. 1994.

AGUIRRE, L. A. **Introdução à identificação de sistemas -Técnicas lineares e não lineares: Teoria e Aplicação**. 4. ed. Belo Horizonte: Editora UFMG, 2015.

AGUIRRE, L. A. **A Bird's Eye View of Nonlinear System Identification**. arXiv preprint arXiv:1907.06803, 2019.

AGUIRRE, L. A.; BILLINGS, S. A. Dynamical effects of overparametrization in nonlinear models. **Physica D: Nonlinear Phenomena**, v. 80, n. 1–2, p. 26–40, 1 jan. 1995.

AGUIRRE, L. A.; DONOSO-GARCIA, P. F.; SANTOS-FILHO, R. Use of a priori information in the identification of global nonlinear models-a case study using a buck converter. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 47, n. 7, p. 1081–1085, jul. 2000.

AGUIRRE, L. A.; RODRIGUES, G. G.; JÁCOME, C. R. F. Identificação de sistemas não lineares utilizando modelos NARMAX polinomiais—uma revisão e novos resultados. **SBA Controle & Automação**, v. 9, n. 2, p. 90–106, 1998.

AHMAD, R.; JAMALUDDIN, H.; HUSSAIN, M. A. Model structure selection for a discrete-time non-linear system using a genetic algorithm. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, v. 218, n. 2, p. 85–98, 2004.

AKAIKE, H. A new look at the statistical model identification. **IEEE Transactions on automatic control**, v. 19, n. 6, p. 716–723, 1974.

AL-DABBAGH, R. D. **Differential evolution with adaptive repository of strategies and parameter control schemes**. 2015 IEEE Congress on Evolutionary Computation (CEC). **Anais...IEEE**, maio 2015

BARROSO, M.; NEPOMUCENO, E. **Multi-objective identification of a buck dc-dc converter**. XV Congresso Brasileiro de Automática. Gramado-RS. **Anais...2004**

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of genetic algorithms: Part 1, fundamentals. **University Computing**, v. 15, n. 2, p. 56–69, 1993.

BIANCHI, F.; FALSONE, A.; PRANDINI, M.; PIRODDI, L. A randomised approach for NARX model identification based on a multivariate Bernoulli distribution. **International Journal of Systems Science**, v. 48, n. 6, p. 1203–1216, 2017.

BILLINGS, S. A. **Nonlinear system identification : NARMAX methods in the time, frequency, and spatio-temporal domains**. Chichester - UK: Wiley, 2013.

BILLINGS, S. A.; CHEN, S.; KORENBERG, M. J. Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. **International Journal of Control**, v. 49, n. 6, p. 2157–2189, 1989.

BILLINGS, S. A.; VOON, W. S. F. Structure detection and model validity tests in the identification of nonlinear systems. **IEE Proceedings D Control Theory and Applications**, v. 130, n. 4, p. 193, 1983.

BILLINGS, S. A.; VOON, W. S. F. Correlation based model validity tests for non-linear models. **International Journal of Control**, v. 44, n. 1, p. 235–244, 23 jul. 1986.

BILLINGS, S. A.; ZHU, Q. M. **Nonlinear model validation using correlation tests**. *International journal of control*, v. 60, n. 6, p. 1107–1120, 1994.

BOX, G. E. P.; JENKINS, G. M. **Time series analysis : forecasting and control**. Holden-Day, 1976.

BREST, J.; GREINER, S.; BOSKOVIC, B.; MERNIK, M.; ZUMER, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. **IEEE transactions on evolutionary computation**, v. 10, n. 6, p. 646–657, 2006.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information sciences**, v. 237, p. 82–117, 2013.

CAMERON, A. C.; WINDMEIJER, F. A. G. R-squared measures for count data regression models with applications to health-care utilization. **Journal of Business & Economic Statistics**, v. 14, n. 2, p. 209–220, 1996.

CHEN, S.; BILLINGS, S. A. Representation of non-linear systems: the NARMAX model. **International Journal of Control**, v. 49, n. 3, p. 303–344, 1989.

CHIANG, C.-L. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. **IEEE transactions on power systems**, v. 20, n. 4, p. 1690–1699, 2005.

CHIANG, C.-W.; LEE, W.-P.; HEH, J.-S. A 2-Opt based differential evolution for global optimization. **Applied Soft Computing**, v. 10, n. 4, p. 1200–1207, 2010.

COELHO, A. A. R.; COELHO, L. DOS S. **Identificação de sistemas dinâmicos lineares**. 2. ed. Florianópolis: Editora da UFSC, 2015.

CONGYING, L.; HUANPING, Z.; XINFENG, Y. **Particle swarm optimization algorithm for quadratic assignment problem**. Proceedings of 2011 International Conference on Computer Science and Network Technology. **Anais...IEEE**, 2011



CORRÊA, M. V. **Identificação de sistemas dinâmicos não-lineares utilizando modelos NARMAX racionais: aplicação a sistemas reais**. 1997. 125 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 1997.

CORRÊA, M. V.; AGUIRRE, L. A. Identificação não-linear caixa-cinza: uma revisão e novos resultados. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, v. 15, n. 2, p. 109–126, jun. 2004.

CORRÊA, M. V.; AGUIRRE, L. A.; SALDANHA, R. R. Using steady-state prior knowledge to constrain parameter estimates in nonlinear system identification. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 49, n. 9, p. 1376–1381, 2002.

CRAWFORD, B.; SOTO, R.; ASTORGA, G.; GARCÍA, J.; CASTRO, C.; PAREDES, F. Putting continuous metaheuristics to work in binary search spaces. **Complexity**, v. 2017, 2017.

DAHI, Z. A. E. M.; MEZIOUD, C.; DRAA, A. **Binary bat algorithm: on the efficiency of mapping functions when handling binary problems using continuous-variable-based metaheuristics**. IFIP International Conference on Computer Science and its Applications. **Anais...**Springer, 2015

DANDY, G. C.; SIMPSON, A. R.; MURPHY, L. J. An improved genetic algorithm for pipe network optimization. **Water resources research**, v. 32, n. 2, p. 449–458, 1996.

DU, K.-L.; SWAMY, M. N. S. Search and optimization by metaheuristics. **Techniques and Algorithms Inspired by Nature**; Birkhauser: Basel, Switzerland, 2016.

FALSONE, A.; PIRODDI, L.; PRANDINI, M. A randomized algorithm for nonlinear model structure selection. **Automatica**, v. 60, p. 227-238, 2015.

FONSECA, C. M.; MENDES, E. M.; FLEMING, P. J.; BILLINGS, S. A. **Non-linear model term selection with genetic algorithms**. Proceedings of IEE/IEEE Workshop on Natural Algorithms in Signal Processing. **Anais...**1993

FONSECA, C. M.; FLEMING, P. J. Non-linear system identification with multiobjective genetic algorithms. **IFAC Proceedings Volumes**, v. 29, n. 1, p. 1169–1174, 1996.

GOLDBARG, E.; GOLDBARG, M.; LUNA, H. **Otimização Combinatória e Metaheurísticas: Algoritmos e aplicações**. Elsevier, 2015.

GOLDBERG, D. E. Genetic algorithms in search. **Optimization and Machine Learning, Reading, Massachusetts**, 1989.

GRAHAM, R. L. **Handbook of combinatorics**. Elsevier, 1995.

GUAN, J. **Differential evolution with a dimensional mutation strategy for global optimization**. 2016 IEEE Congress on Evolutionary Computation (CEC). Vancouver, BC, Canada. **Anais...**IEEE, jul. 2016

HAFIZ, F.; SWAIN, A.; MENDES, E. M.; PATEL, N. **Structure selection of polynomial NARX models using two dimensional (2D) particle swarms**. 2018 IEEE Congress on Evolutionary Computation (CEC). **Anais...**IEEE, 2018

HAFIZ, F.; SWAIN, A.; MENDES, E. M. A. M. Two-Dimensional (2D) particle swarms for structure selection of nonlinear systems. **Neurocomputing**, v. 367, p. 114–129, 2019.

HAUPT, R. L.; HAUPT, S. E. **Practical genetic algorithms**. New Jersey: Wiley-Interscience, 2004.

HAYKIN, S. **Redes Neurais: Princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001.

HO, C. K. S.; FRENCH, I. G.; COX, C. S.; FLETCHER, I. **Genetic algorithms in structure identification for NARX models**. Artificial Neural Nets and Genetic Algorithms. **Anais...**Springer, 1998

HOLLAND, J. **Adaptation in natural and artificial systems: an introductory analysis with application to biology, control and artificial intelligence**. MIT Press, 1975.

HUNG, N. T.; ISMAIL, I.; SAAD, N. B.; TUFA, L.; IRFAN, M. **Design of optimal GBN sequences for identification of MIMO systems**. 2015 10th Asian Control Conference (ASCC). **Anais...**IEEE, 2015

ISERMANN, R.; MÜNCHHOF, M. **Identification of Dynamic Systems**. 1. ed. Heidelberg: Springer, 2011.

JAMIL, M.; ZEPERNIC, H. J.; YANG, X. S. **Improved bat algorithm for global optimization**. Applied Soft Computing, p. 49–72, 2013.

JUANG, C.-F.; HSIEH, C.-D. A locally recurrent fuzzy neural network with support vector regression for dynamic-system modeling. **IEEE Transactions on Fuzzy Systems**, v. 18, n. 2, p. 261–273, 2010.

KADTKE, J. B.; BRUSH, J.; HOLZFUSS, J. Global Dynamical Equations and Lyapunov Exponents from Noisy Chaotic Time Series. **International Journal of Bifurcation and Chaos**, v. 03, n. 03, p. 607–616, 20 jun. 1993.

KENNEDY, J.; EBERHART, R. **Particle swarm optimization**. Proceedings of ICNN'95-International Conference on Neural Networks. **Anais...**IEEE, 1995

KENNEDY, J.; EBERHART, R. C. **A discrete binary version of the particle swarm algorithm**. 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation. **Anais...**IEEE, 1997

KHAN, K.; NIKOV, A.; SAHAI, A. **A fuzzy bat clustering method for ergonomic screening of office workplaces**. Third International Conference on Software, Services and Semantic Technologies S3T 2011. Anais...Springer, 2011

KORENBERG, M.; BILLINGS, S. A.; LIU, Y. P.; MCILROY, P. J. Orthogonal parameter estimation algorithm for non-linear stochastic systems. **International Journal of Control**, v. 48, n. 1, p. 193–210, 29 jul. 1988.

KUMON, T.; IWASAKI, M.; SUZUKI, T.; HASHIYAMA, T.; MATSUI, N.; OKUMA, S. **Nonlinear system identification using genetic algorithm**. 2000 26th Annual Conference of the IEEE Industrial Electronics Society. **Anais...IEEE**, 2000

LEONTARITIS, I. J.; BILLINGS, S. A. Input-output parametric models for non-linear systems Part I: deterministic non-linear systems. **International Journal of Control**, v. 41, n. 2, p. 303–328, 21 fev. 1985a.

LEONTARITIS, I. J.; BILLINGS, S. A. Input-output parametric models for non-linear systems Part II: stochastic non-linear systems. **International Journal of Control**, v. 41, n. 2, p. 329–344, 21 fev. 1985b.

LEUNG, F. H. F.; LAM, H. K.; LING, S. H.; TAM, P. K. S. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. **IEEE Transactions on Neural networks**, v. 14, n. 1, p. 79–88, 2003.

LIN, J. H.; CHOU, C. W., YANG, C. H., & TSAI, H. L. **A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems**. Computer and Information Technology, v. 2, n. 2, p. 56–63, 2012.

LIU, D.; TENG, H. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. **European Journal of Operational Research**, v. 112, n. 2, p. 413–420, 1999.

LIU, G. P. **Nonlinear Identification and Control: A Neural Network Approach**. London: Springer London, 2001.

LIU, J.; LAMPINEN, J. A fuzzy adaptive differential evolution algorithm. **Soft Computing**, v. 9, n. 6, p. 448–462, 2005.

LUH, G.-C.; WU, C.-Y. Non-linear system identification using genetic algorithms. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, v. 213, n. 2, p. 105–118, 27 mar. 1999.

MADÁR, J.; ABONYI, J.; SZEIFERT, F. Genetic programming for the identification of nonlinear input– output models. **Industrial & engineering chemistry research**, v. 44, n. 9, p. 3178–3186, 2005.

MARTINS, S. A. M.; NEPOMUCENO, E. G.; BARROSO, M. F. S. Improved structure detection for polynomial NARX models using a multiobjective error reduction ratio. **Journal of Control, Automation and Electrical Systems**, v. 24, n. 6, p. 764–772, 2013.

MITCHELL, M. **An introduction to genetic algorithms**. MIT press, 1998.

NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **IEEE Transactions on Neural Networks**, v. 1, n. 1, p. 4–27, mar. 1990.

NELLES, O. **Nonlinear system identification: from classical approaches to neural networks and fuzzy models**. Springer Science & Business Media, 2000.

OLIVEIRA, Á. H. R.; LEANDRO, G. V. Algoritmos de Otimização Combinados Aplicados à Identificação de Sistemas. **XIV Conferência Brasileira de Dinâmica, Controle e Aplicações**, 2019.

PAMPARA, G.; FRANKEN, N.; ENGELBRECHT, A. P. **Combining particle swarm optimisation with angle modulation to solve binary problems**. 2005 IEEE congress on evolutionary computation. **Anais...IEEE**, 2005

PARK, J. B.; PARK, Y. M.; WON, J. R.; LEE, K. Y. An improved genetic algorithm for generation expansion planning. **IEEE Transactions on Power Systems**, v. 15, n. 3, p. 916–922, 2000.

PEÑARANDA, J. R. C.; SAAVEDRA-MONTES, A. J. **Dynamic model validation via error indexes**. 2012 IEEE 4th Colombian Workshop on Circuits and Systems (CWCAS). **Anais...IEEE**, 2012

PHAM, D. TRUONG.; LIU, XING. **Neural Networks for Identification, Prediction and Control**. Springer London, 1995.

PRICE, K. V. **Differential evolution: a fast and simple numerical optimizer**. Proceedings of North American Fuzzy Information Processing. **Anais...IEEE**, 1996

PRICE, K. V.; STORN, R. M.; LAMPINEN, J. A. **Differential evolution : a practical approach to global optimization**. Springer, 2005.

PROAKIS, J. G.; SALEHI, M.; ZHOU, N.; LI, X. **Communication systems engineering**. 2. ed. Prentice Hall New Jersey, 2002.

QIN, A. K.; SUGANTHAN, P. N. **Self-adaptive differential evolution algorithm for numerical optimization**. 2005 IEEE congress on evolutionary computation. **Anais...IEEE**, 2005

RETES, P. F. L.; AGUIRRE, L. A. NARMAX model identification using a randomised approach. **International Journal of Modelling, Identification and Control**, v. 31, n. 3, p. 205, 2019.

RODRIGUES, G. G. **Identificação de sistemas dinamicos não-lineares utilizando modelos narmax polinomiais: aplicação a sistemas reais**. 1996. 127 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Minas Gerais, Belo Horizonte, 1996.

SCHETZEN, M. **The Volterra and Wiener theories of nonlinear systems**. 1980.

SEVERINO, A. G. V.; ARAÚJO, F. M. U. DE. Meta-Heurísticas Aplicadas à Identificação de Sistemas. **XIII Simpósio Brasileiro de Automação Inteligente**, 2017.

SIARRY, P. **Metaheuristics**. Cham: Springer International Publishing, 2016.

SILVA, E. L. DA; GIL, H. A.; AREIZA, J. M. Transmission network expansion planning under an improved genetic algorithm. **IEEE Transactions on Power Systems**, v. 15, n. 3, p. 1168–1174, 2000.

SÖRENSEN, K.; GLOVER, F. Metaheuristics. **Encyclopedia of operations research and management science**, v. 62, p. 960–970, 2013.

SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A survey. **computer**, v. 27, n. 6, p. 17–26, 1994.

STORN, R.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997.

TALMON, R.; KUSHNIR, D.; COIFMAN, R. R.; COHEN, I.; GANNOT, S. Parametrization of linear systems using diffusion kernels. **IEEE Transactions on Signal Processing**, v. 60, n. 3, p. 1159–1173, 2011.

THOUVEREZ, F.; JEZEQUEL, L. Identification of NARMAX models on a modal base. **Journal of Sound and Vibration**, v. 189, n. 2, p. 193–213, 1996.

VAILLANT, O. R.; KURAMOTO, A. S. R.; GARCIA, C. Effectiveness of signal excitation design methods for identification of ill-conditioned and highly interactive processes. **Industrial & Engineering Chemistry Research**, v. 52, n. 14, p. 5120–5135, 2013.

WANG, L.; CLUETT, W. R. Use of PRESS residuals in dynamic system identification. **Automatica**, v. 32, n. 5, p. 781–784, 1996.

WANG, Y.; CAI, Z.; ZHANG, Q. Enhancing the search ability of differential evolution through orthogonal crossover. **Information Sciences**, v. 185, n. 1, p. 153–177, 2012.

XIE, J.; ZHOU, Y.; CHEN, H. **A novel bat algorithm based on differential operator and Lévy flights trajectory**. Computational intelligence and neuroscience, v. 2013, 2013.

YANG, X.-S. **A New Metaheuristic Bat-Inspired Algorithm**. Nature inspired cooperative strategies for optimization (NICSO 2010), p. 65–74, 2010.

YANG, X.-S. **Bat algorithm for multi-objective optimisation**. International Journal of Bio-Inspired Computation, v. 3, n. 5, p. 267–274, 2011.

YANG, X.-S.; HE, X. **Bat algorithm: literature review and applications**. International Journal of Bio-inspired computation, v. 5, n. 3, p. 141–149, 2013.

YANG, X.; GANDOMI, A. H. **Bat algorithm: a novel approach for global engineering optimization**. Engineering computations, 2012.

YASSIN, I. M.; TAIB, M. N.; ADNAN, R. Recent advancements & methodologies in system identification: A review. **Scientific Research Journal**, v. 1, n. 1, p. 14–33, 2013.

ZAKARIA, M. Z.; NOR, A. M.; JAMALUDDIN, H.; AHMAD, R. Multi-Objective Optimization Using Differential Evolution for Modeling Automotive Palm Oil Biodiesel Engine. **WSEAS Trans. Syst. Control**, v. 9, n. 1, p. 500–513, 2014.

ZAKARIA, M. Z.; SAAD, M. S.; JAMALUDDIN, H.; AHMAD, R.. **Dynamic system modeling of flexible beam system using multi-objective optimization differential evolution algorithm**. Applied Mechanics and Materials. **Anais...Trans Tech Publ**, 2015



ZAKARIA, M. Z.; MANSOR, Z.; NOR, A. M.; SAAD, M. S.; BAHARUDIN, M. E.; AHMAD, R. NARMAX Model Identification Using Multi-Objective Optimization Differential Evolution. **International Journal of Integrated Engineering**, v. 10, n. 7, 2018.

ZHANG, J; SANDERSON, A. C. **JADE: Self-adaptive differential evolution with fast and reliable convergence performance**. 2007 IEEE Congress on Evolutionary Computation. **Anais...IEEE**, set. 2007

ZHOU, X. G.; ZHANG, G. J.; HAO, X. H.; YU, L.; XU, D. W. **Differential evolution with multi-stage strategies for global optimization**. 2016 IEEE Congress on Evolutionary Computation (CEC). **Anais...IEEE**, jul. 2016

ZHU, Y. **Multivariable system identification for process control**. Elsevier, 2001.